Smith, Jonathan Mark (1988) The automated analysis and design of underground concrete nuclear shelters. Masters thesis, Middlesex Polytechnic. [Thesis]

Final accepted version (with author's formatting)

Smith, Jonathan Mark, 1988.
The automated analysis and deign of underground concrete nuclear shelters.
Available from Middlesex University's Research Repository.

THE AUTOMATED ANALYSIS AND DESIGN OF UNDERGROUND CONCRETE NUCLEAR SHELTERS

A thesis submitted to the

COUNCIL FOR NATIONAL ACADEMIC AWARDS

for the degree of

MASTER OF PHILOSOPHY

by

Jonathan Mark Smith B.Sc.

SCHOOL OF CIVIL ENGINEERING

MIDDLESEX POLYTECHNIC, LONDON

June 1988

ABSTRACT

Two topics are covered in this thesis; the implementation of the SemiLoof shell element into ANSYS and a finite element analysis of an underground nuclear shelter subjected to a blast load, using SemiLoof shells.
The SemiLoof Shell is an eight noded, isoparametric, thin shell element. Its main advantage over its competitors is it's faster matrix calculation and solution time due mainly to the fact that it has only 32 degrees of freedom (3 translationals at each node, plus 2 rotationals at each midside node), rather than the usual 48 degrees of freedom (6 at each node).

The User Element Utility was used to link the original SemiLoof Fortran subroutines into ANSYS as a linear element. As these subroutines were already available, the main task was to design and write interfacing subroutines which were required to translate parametric and array names, serve subroutine calling arguments and overcome inconsistencies in the arrangement of the ANSYS/SemiLoof degree of freedom sets. The original SemiLoof calculations are limited to displacements, element forces and bending moments, hence further stress calculation subroutines had to be written in order to make the element of practical use to the engineer. SemiLoof is referenced by ANSYS as Stif100 can be used in the same way as any other ANSYS element. All the linear analysis types, pre and postprocessing features and wave front solution are available for use by SemiLoof, which can also be mixed with other ANSYS elements having a suitable degree of freedom set. Benchmark tests on SemiLoof in ANSYS have shown it to be a quick and accurate performer and a useful alternative to ANSYS's eight node thin shell - Stif93.

SemiLoof in ANSYS was then used to analyse an underground, reinforced concrete nuclear shelter, subjected to an impulsive blast load. The aim of this analysis was to determine if the design data - bending moments - were of a realistic magnitude. The SemiLoof analyses showed that they were less than adequate and that the shelter is probably slightly underdesigned.
It suggested in this thesis that the design criteria of moderate damage was unsuitable for the purposes of the shelter and that it would be advantageous to design the shelter so that it would remain in good condition after the blast load, and thereby service the basic needs of comfort and hygiene for the occupants.

CONTENTS Page No.

ACKNOWLEDGEMENTS

Many thanks to Dr Y. Bangash of the Dept. of Civil Engineering at Middlesex Polytechnic, for initiating the research and providing guidance and support throughout the research period.

Many thanks also to Dr P.T.K. Lim of Structures and Computers Ltd, for constructive help and stimulating discussion on some of the technical aspects of the work. Structures and Computers also generously made available the necessary hardware and software items which were required for this thesis.

Finally, thanks for the tolerance and support shown by family, friends and collegues throughout the duration of this project.

NOTATION

Coordinate systems

X, Y, Z	Global cartesian coordinates
x, y, z	Nodal cartesian coordinates
θx, θy, θz	Nodal rotations
θa, θb	Loof rotations
ξ, η	Element curvilinear coordinate system

Element matrices

[D]	Elasticity matrix
[B]	Shape function derivitive matrix
[K]	Stiffness matrix
[M]	Mass matrix
[C]	Damping matrix
{F}	Nodal force vector
{u}	Nodal displacement vector
$\{\dot{u}\}$	Nodal velocity vector
$\{\ddot{u}\}$	Nodal acceleration vector
[J]	Jacobian transformation matrix
det J	Jacobian determinant

GENERAL INTRODUCTION

In 1973 Professor Bruce Irons of the University College Swansea, presented two new displacement based structural isoparametric finite elements called the SemiLoof Beam and SemiLoof Shell,(3,4). Since this initial publication, a number of researchers have enhanced the elements in order to include nonlinearities, anisotropic material properties, varying nodal thicknesses and a heat transfer capability.

Several major finite element codes (e.g. LUSAS, BERSAFE, (5,6)) have already implemented their own versions of the SemiLoof elements, though the general purpose program ANSYS, currently has no Semiloof elements in its library.

A new feature of ANSYS is its ability to interface with a user supplied element. The user supplies the matrix calculation subroutines, which can then be linked with the main program, enabling the user to take advantage of ANSYS's extensive capabilities of pre and post processsing, and solution routines.

The ANSYS user element utility is documented for the first time in Appendix A, and explains the design of the utility and its interaction with the main program ANSYS.

The implementation itself requires a number of interfacing operations to overcome the incompatibilities between the two existing ANSYS and SemiLoof codes, and all the stress items have been calculated by new coding. A detailed description of the incompatibilities, solutions and stress calculations are fully documented in Chapter 1 and their computer coding presented in Appendix A.

The implementation of SemiLoof into ANSYS forms the main and original part of this thesis.

In the second part of this thesis, SemiLoof shells are used in a transient analysis of a semi-buried, reinforced concrete nuclear shelter.

As there can be no guarantees with a nuclear shelter, it is important to establish that it has been designed with sensible criteria and realistic loading data. The Home Office's recommended design for a domestic, semi-buried reinforced concrete nuclear shelter is evaluated in this thesis, by checking both the design criteria and the distribution and magnitude of loading that might arise from a nuclear detonation. The design data is verified by repeating the original Home Office analyses with a more refined finite element model using SemiLoof shells. Recommendataions for the review of the design philosophy are also presented.

CHAPTER 1

THE IMPLEMENTATION OF THE SEMILOOF SHELL ELEMENT INTO ANSYS

1.1 Introduction

ANSYS is a general purpose finite element program for engineering analysis and includes preprocessing, solution and postprocessing capabilities supported by extensive graphics.

The ANSYS element library contains over 70 elements, several of which are structural shells:

Stif41	Membrane shell
Stif48	Plastic triangular shell
Stif53	Triangular laminated shell
Stif63	Four node quadrilateral shell
Stif91	Eight node layered shell
Stif93	Eight node isoparametric shell
Stif99	Eight node layered shell

For a simple shell analysis in ANSYS, the elements most likely to be used would be either a Stif63 or Stif93. The addition of the SemiLoof will provide a further option.

A User Element Utility is available in ANSYS and comes in the form of a standard set of Fortran 77 subroutines which may be modified or replaced by the user's own routines. ANSYS Revision 4.3,(1), will be used in this thesis to host the SemiLoof shell element.

1.2 The SemiLoof shell element

The SemiLoof shell element is an eight noded thin structural shell element. It differs from most structural shells by virtue of the fact that the rotations are calculated along its edges rather than at its corners. This arrangements allows the SemiLoof shell to acheive good bending charecteristics with 33% fewer degrees of freedom than its traditional eight noded thin shell counterparts, thus acheiving a faster element formation and solution, without loss of accuracy. The ANSYS User Element utility will be used to include the SemiLoof shell element into the ANSYS element library.

The basis of the SemiLoof shell rotations theory is derived from a degenerated solid element. Two sets of mapping functions describe the element, which is 'Isoparametric', i.e. the same shape functions which describe the geometry also describe the displacement responses. The nodal degree of freedom set is shown in Figs.1.1 and 1.6. Although the final configuration of the stiffness matrix contains only 32 equations, 43 equations contribute to the shell formulation, 11 of which are constrained in order to moderate the element flexibility.

SemiLoof uses standard eight node quadarilateral interpolation equations, derived from a polynomial of eight terms as shown in Table 1.1, to map both the mid plane geometry and the translational responses. A typical membrane shape function, evaluated at a corner node, is shown in Fig.1.3.

A further term is added, evaluated at the element centre, in each orthogonal direction (a bubble function as shown in Fig.1.4.) to allow the element to pass the quaderilateral patch test.

The rotational responses and locations on the top and bottom surfaces of the element are mapped using a second set of shape functions, derived from the same polynomial series as the membrane

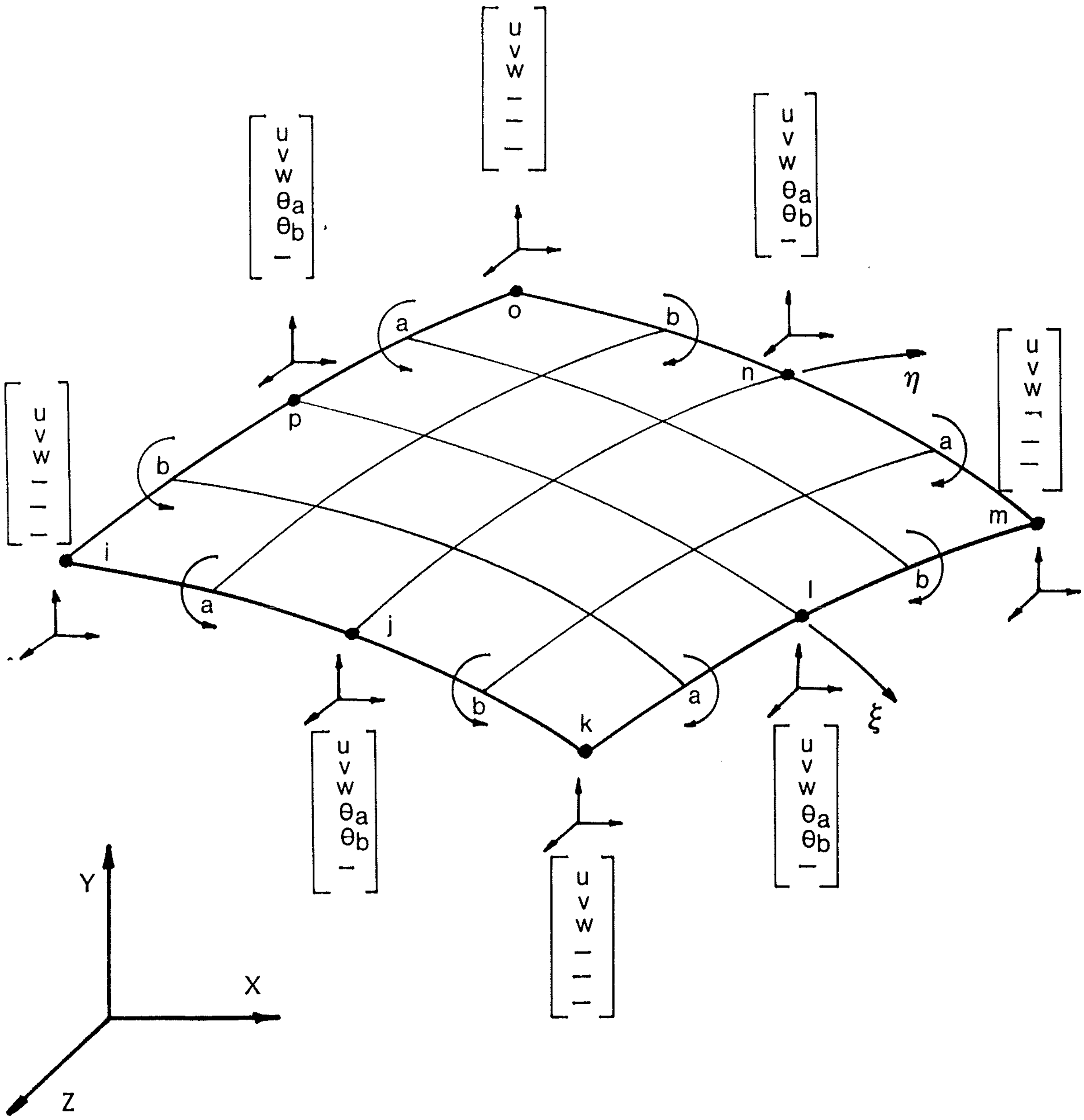


Figure 1.1 The SemiLoof shell element.

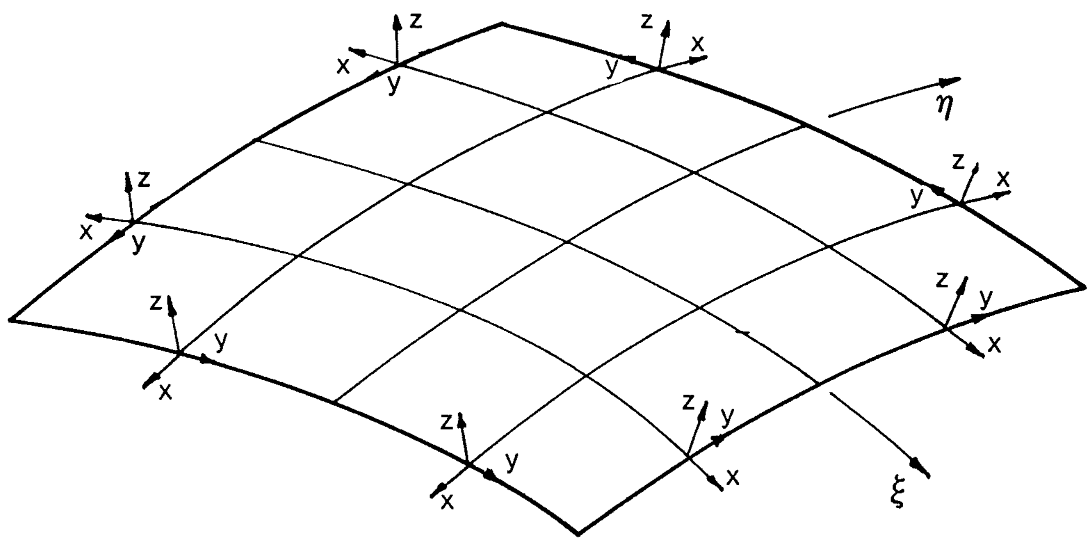


Figure 1.2 Loof nodal degrees of freedom.

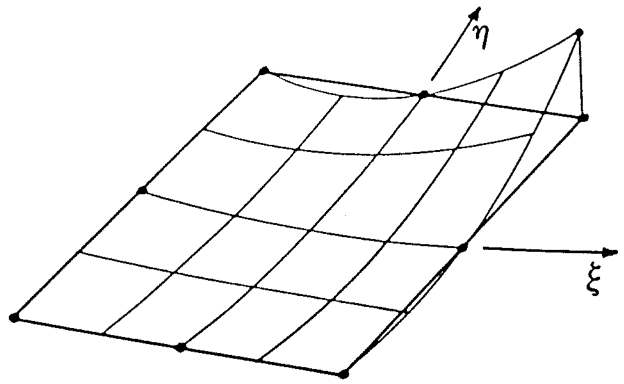


Figure 1.3 Membrane mapping function

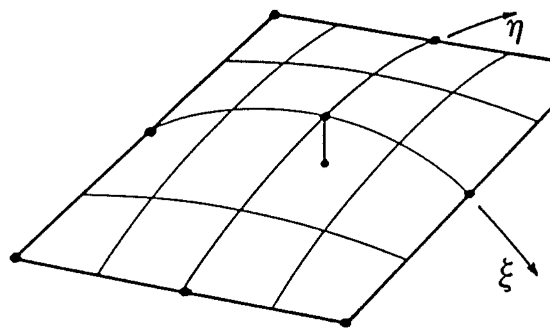


Figure 1.4 Bubble function

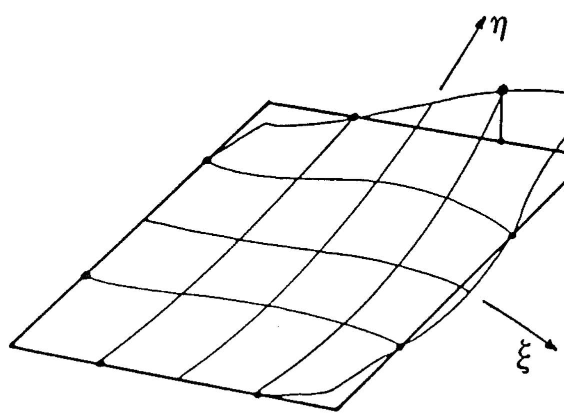


Figure 1.5 Loof mapping function

functions, but with one extra, ninth term (Table 1.2). These polynomials are evaluated at the Loof and central nodes. (See Fig.1.2). A typical 'Loof' node mapping function is shown in Fig.1.5. The rotations are mapped by first setting up orthogonal vector systems at the user defined corner and midside nodes, and then interpolating them at the Loof nodes using the membrane mapping functions.

Once at the Loof nodes , the vectors are then mapped within the element, using the Loof shape functions, for later stress calculations.

Numerical integration is carried out at the standard 2x2 Gauss points for the work done by the membrane responses. The work done by the rotations is calculated at the Loof nodes.

The element 'stresses' (i.e. Force per m/run and moment per m/run) are calculated at the same Gauss points within the element which were used as sampling points for the numerical integration of the translational stiffness. Both membrane and rotational responses are mapped to those points but by using their respective interpolation equations.

In total there are 43 equations (24 translational, 1 bubble function and 18 rotational terms), however, each rotation about the local x axis at the Loof nodes (see Fig.1.2) is constrained out together with the two central rotations, leaving 33 degrees of freedom. The one remaining degree of freedom at the central node (bubble function) is constrained out leaving 32 degrees of freedom as shown in Fig.1.1.

A definitive account of the SemiLoof shell formulation is given in ref.(4).

Table 1.1 Polynomials series to derive translational mapping functions evaluated at +- 1

$$f(x,y,z) = a1 + a2x + a3y + a4x^2 + a5xy + a6y^2 + a7x^2y + a8xy^2$$

Translational Shape functions

$N1 = 1/4\ (1-\xi)\ (1-\eta)\ (-\xi-\eta-1)$

$N2 = 1/2\ (1-\xi^2)\ (1-\eta)$

$N3 = 1/4\ (1+\xi)\ (1-\eta)\ (\xi-\eta-1)$

$N4 = 1/2\ (1+\xi^2)\ (1-\eta^2)$

$N5 = 1/4\ (1+\xi)\ (1+\eta)\ (\xi+\eta-1)$

$N6 = 1/2\ (1-\xi^2)\ (1+\eta)$

$N7 = 1/4\ (1-\xi)\ (1+\eta)\ (-\xi+\eta-1)$

$N8 = 1/2\ (1-\xi^2)\ (1-\eta^2)$

$N9 = (1-\xi^2)(1-\eta^2)$ Bubble function

Table 1.2 Polynomials series to derive rotational mapping functions evaluated at +- 1

$$f(01,02) = a1 + a2x + a3y + a4x^2 + a5xy + a6y^2 + a7x^2y + a8xy^2$$

$$+ \xi\eta\cancel{(\xi^2-\eta^2)}$$

$$xy(x^2-y^2)$$

Rotational Shape functions

$$L1 = 3/32(3\xi^2-\eta^2) + 1/8[(-3\xi(1-\xi^2) - \mathrm{sqrt}(3).\xi(3\xi^2-\eta-1-3/2\eta(\eta^2-\xi^2))]$$

$$L2 = 3/32(3\xi^2-\eta^2) + 1/8[(-3\xi(1-\xi^2) + \mathrm{sqrt}(3).\xi(3\xi^2-\eta-1-3/2\eta(\eta^2-\xi^2))]$$

$$L3 = 3/32(3\xi^2-\eta^2) + 1/8[(+3\xi(1-\xi^2) + \mathrm{sqrt}(3).\xi(3\xi^2-\xi-1-3/2\xi(\xi^2-\eta^2))]$$

$$L4 = 3/32(3\xi^2-\eta^2) + 1/8[(+3\xi(1-\xi^2) + \mathrm{sqrt}(3).\xi(3\xi^2+\xi-1+3/2\xi(\xi^2-\eta^2))]$$

$$L5 = 3/32(3\xi^2-\eta^2) + 1/8[(+3\xi(1-\xi^2) + \mathrm{sqrt}(3).\xi(3\xi^2+\eta-1+3/2\eta(\eta^2-\xi^2))]$$

$$L6 = 3/32(3\xi^2-\eta^2) + 1/8[(+3\xi(1-\xi^2) - \mathrm{sqrt}(3).\xi(3\xi^2+\eta-1+3/2\eta(\eta^2-\xi^2))]$$

$$L7 = 3/32(3\xi^2-\eta^2) + 1/8[(+3\xi(1-\xi^2) - \mathrm{sqrt}(3).\xi(3\xi^2+\xi-1+3/2\xi(\xi^2-\eta^2))]$$

$$L8 = 3/32(3\xi^2-\eta^2) + 1/8[(-3\xi(1-\xi^2) - \mathrm{sqrt}(3).\xi(3\xi^2-\xi-1-3/2\xi(\xi^2-\eta^2))]$$

1.3 The SemiLoof shell subroutines

The coding for a complete finite element program is presented in (4). and contains in its library, subroutines for the production of SemiLoof beam and shell elements. Approximately twelve subroutines work under the control of the full program and of these, six are used to generate the SemiLoof shell stiffness matrix.

After identification of the individual roles of the subroutines, those which appeared to contribute to the generation of the stiffness matrix were extracted and interrogated to identify their full scope within the program.

An elementary data input program and a matrix solver were written and linked with the SemiLoof subroutines and run as a stand-alone single element type finite element program, as illustrated in Table 1.3. The coding for the stand-alone program and the results of a simple examples are documented in Appendix C.

After further debugging and modification of the SemiLoof subroutines, satisfactory displacement results were obtained.

Five subroutines were extracted, requiring only minor modifications:

derivitives

HALOOF - Used to calculate the shape function array [B].

SFR - Supplies the shape functions and their derivitives to HALOOF.

SCALAR - A service subroutine which calculates a scalar product.

VECTOR - A service subroutine which calculates a vector product.

BLOCK DATA - Contains initialisation coefficients and common blocks.

Table 1.3   Stand-alone Semiloof shell element program for the debugging and verification of the extracted subroutines.

```
|-----------|              Input data defined for one element
| DATA PREP |              model: Nodes
|-----------|                     Nodal coordinates
      |                           Element definitions
      |                           Material properties
      |
      |
      |
|-----------|              Data recieved and placed into arrays
|   LOOF    |              to be passed to HALOOF.
|-----------|
      |
      |        |--------|
      |        |  SFR   |
|-----------|--|--------|  Matrix of shape function derivitives
|  HALOOF   |--| SCALAR |  [B] calculated and passed back to LOOF.
|-----------|--|--------|
      |        | VECTOR |
      |        |--------|
      |
      |
| ----------|              Calculation of elasticity matrix [D].
|   LOOF    |              Calculation of stiffness matrix [K].
|-----------|              [K] matrix passed back to main program
      |                    for solution.
      |
      |
      |
      |        |--------|
|-----------|  | FORCE  |  Force vector {F} defined, displacement
| EQUATION  |--| VECTOR |  constraints added to [K].
| SOLUTION  |  |+B.C.'s|  Equation {F}=[K]*{u} solved.
|-----------|  |--------|
      |
      |
      |
|---------------|
|     PRINT     |          Verification of displacement results.
| DISPLACEMENTS |
|---------------|
```

Table 1.3 Stand-alone Semiloof shell element program for the debugging and verification of the extracted subroutines.

```
|-----------|                Input data defined for one element
| DATA PREP |                model: Nodes
|-----------|                       Nodal coordinates
      |                             Element definitions
      |                             Material properties
      |
      |
      |
|-----------|                Data recieved and placed into arrays
|   LOOF    |                to be passed to HALOOF.
|-----------|
      |
      |        |--------|
      |        |  SFR   |
|-----------|--|--------| Matrix of shape function derivitives
|  HALOOF   |--| SCALAR | [B] calculated and passed back to LOOF.
|-----------|--|--------|
      |        | VECTOR |
      |        |--------|
      |
      |
| ----------|                Calculation of elasticity matrix [D].
|   LOOF    |                Calculation of stiffness matrix [K].
|-----------|                [K] matrix passed back to main program
      |                      for solution.
      |
      |
      |
      |        |--------|
|-----------|  | FORCE  | Force vector {F} defined, displacement
| EQUATION  |--| VECTOR | constraints added to [K].
| SOLUTION  |  | +B.C.'s| Equation {F}=[K]*{u} solved.
|-----------|  |--------|
      |
      |
      |
|---------------|
|     PRINT     |            Verification of displacement results.
| DISPLACEMENTS |
|---------------|
```

One other subroutine, requiring more extensive modification due to its involvement at the interface stage of the process, was SHELL.

SHELL - Recieves data from the main program and prepares further data to pass to HALOOF.

Calculates the elasticity matrix [D].

The shape function derivitives [B], are passed from HALOOF and the stiffness matrix calculation:

$[K] = \int_A ([B]^t * [D] * [B])\, dA$, is performed.

The stiffness matrix [K], is passed back to the main program.

Subroutine SHELL was partly re-written to facilitate the interface with the stand-alone program and renamed LOOF. It still essentially performed its original function as specified above.
Listings of the SemiLoof subroutines are given in Appendix A.
Having obtained satisfactory displacement solutions from the elementary SemiLoof finite element program, the requirements for the interfacing of the SemiLoof subroutines with ANSYS were investigated.

1.4 Interfacing the SemiLoof shell with ANSYS

The scope of the User Element Utility is such that it allows users to calculate within ANSYS, their own stiffness, mass and damping matrices and geometry dependant force vectors, whilst still taking advantage of other calculation and data presentation capabilities offered by ANSYS.

Table 1.4 shows the role of the User Element Utility in a typical linear static finite element procedure within ANSYS.
The User Element Utility is detailed in Appendix A.

The tasks facing the ANSYS user element programmer are:

a. Produce a set of subroutines which will calculate the required elemental matrices suitable for use within ANSYS.
b. Design a further set of subroutines which will serve to interface the programming logic, parametric and array names used in both ANSYS and the user element subroutines. The interface subroutines will front and back end the user element subroutines.
c. Write subroutines to calculate the appropriate post data items (e.g. Stresses).
d. Compile and link the above subroutines with ANSYS.
e. Verify the results of the calculations to the satisfaction of those concerned.

The user element subroutines are included on the installation tape with the main program ANSYS, in a program file called USER.ROUTINES.
In order to implement a user defined element, these subroutines must be modified by the user and linked with ANSYS.
There are a total of four user element subroutines supplied to the user for modification:

a. USEREL - Defines user element parameters.
b. USERPT - Defines user element plot shape.
c. ST100 - Contains the coding for the production of element matrices.

Table 1.4 Schematic illustration of the scope of the User Element Utility in ANSYS.

```
                |--------------------------|
                | PREPROCESSING            |
                |                          |
                | Node definition          |
                | Element definition       |
                | Material properties      |
                | Boundary conditions      |
                | Applied loads            |
                |--------------------------|
                             |
                             |
                             |
        |---------------------------------------------------|
        | SOLUTION                                          |
        |                                                   |
  ***   | Element stiffness matrix calculation              |  ***
        | Global stiffness matrix assembly and reduction    |
        | Calculation of displacements                      |
  ***   | Calculation of stresses                           |  ***
        | Calculation of reaction forces                    |
        |---------------------------------------------------|
                             |
                             |
                             |
                |---------------------------|
                | POSTPROCESSING            |
                |                           |
                | Presentation of results   |
                |---------------------------|
```

d. SR100 - Contains coding for the calculation of stresses.

Subroutines USEREL and USERPT are used for pre and postprocessing. ST100 is used in the displacement solution phase for the generation of element matrices.
SR100 is used in the stress pass for calculation of stresses and post displacement items.
After linking, the user element has effectively been added to the ANSYS element library and is referenced by ANSYS as Stif100.
Thus when defining the model geometry in PREP7, the user simply defines element type 100 as one would any other element type and uses it the analysis in the normal way. All the appropriate capabilities of ANSYS are available for use with the user element.
Any element used in ANSYS will have some involvement in each of the three stages of the finite element analyis; those of preprocessing, solution and postprocessing. Any or all of the three phases can be conducted in an interactive or batch processing environment.
Typical steps in a finite element analysis with ANSYS would be:

a. Create the analysis input file, FILE27, with the preprocessor PREP7.
b. Solve the analysis.
c. Evaluate the results with the postprocessor, POST1.

This process, together with ANSYS routines and files is shown in Table 1.5.

Table 1.5 Schematic illustration of the role of the preprocessor, solution and postprocessor in ANSYS together with primary files created.

INPUT FROM KEYBOARD OR BATCH FILE	Data is input into PREP7 from the keyboard or a previously prepared batch file.
PREP7	PREP7 recieves the data input, including the element type definition, carries out checks for completeness and writes FILE27.
FILE27	FILE27 is a condensed, formatted file containing all the information about the finite element model required for its solution.
SOLUTION	The solution phase reads FILE27, and organises the analysis. The solution proceeds until halted by numerical errors or end of data.
FILE12	FILE12 is written by the solution phase and contains both the results of the solution and much of the original model data.
POST1	POST1 reads FILE12 and presents the datain a graphical or tabular format as requested by the user.

Although the capability exists to define user mass and damping matrices for use in structural dynamics analyses, only the SemiLoof stiffness matrix and corresponding force vector calculation options have been utilised in this thesis.

This implies that when using the SemiLoof shell for dynamic analyses, any structure mass must be defined at the nodes by using discrete mass elements. The damping capability will similarly be limited to discrete viscous dashpots and Rayleigh damping constants, (1).

Certain incompatabilities were found to exist between the coding for the SemiLoof shell and the ANSYS User Element Utility. This section now details the incompatibilties and reviews the changes and additional programming required to effect an interface between the two codes.

Two additional subroutines were written to accomodate the interfacing logic:

INTERIN - Modifies data from FILE27 before the SemiLoof subroutines are encountered.

INTEROUT - Modifies the matrices created by the SemiLoof routines before they are passed to the solution phase.

The elementary stand-alone SemiLoof program discussed in the previous section was stripped of its data input and solution routines and the feasibility of linking directly with ANSYS was examined.

The incompatibilities between the ANSYS and SemiLoof codings were identified as follows:

1.4.1. Inconsistent Nodal Degrees of Freedom.

When defining a user element to ANSYS, the user must define the element degree of freedom set in the ANSYS subroutine, USEREL. However, the nodal degree of freedom choice is restricted to that given in the USEREL menu (see Appendix A). The most suitable degree of freedom stencil available appeared to be the six degree of freedom per node set.

SemiLoof has sixteen nodes (including the Loof nodes), bringing the total number of degrees of freedom, assuming a six degree of freedom per node stencil, to ninety-six. Attempting to use this degree of freedom arrangement would be wasteful in terms of array storage but is anyway beyond the limits of the total number of degrees of freedom allowed for a user element, which is sixty.

It was decided to explore the possibility of using an eight noded shell to accomodate the thirty-two degrees of freedom offered by SemiLoof. This has been the nodal degree of freedom arrangement adopted previously by other SemiLoof shell programmers (see Fig.1.7). As no nodes are positioned at the Loof points in the standard eight node ANSYS stencil, the degrees of freedom at the Loof nodes would need to be referenced by the ANSYS element stencil at either the midside or corner nodes. By defining the Loof rotations at the midside nodes, the nodal degree of freedom set seen by PREP7 would be (u_x,u_y,u_z,θ_x,θ_y,θ_z) at each of the eight nodes of the element. However the true degree of freedom set, contained within that stencil is as shown in Fig.1.7.

The result is that although the forty-eight degree of freedom set seen by ANSYS is adequate to define the SemiLoof degree of freedom set, all of the corner rotations are null and those at the midsides are not truly representative of the SemiLoof geometric locations or interpretations.

```
ux uy uz        θx     ux uy uz     θx       ux uy uz
        * ------ * ------- * ------- * ------ *
        ¦                                     ¦
        ¦                                     ¦
        ¦                                     ¦
     θy *                  Y                  * θy
        ¦                  ¦                  ¦
        ¦                  ¦                  ¦
        ¦                  ¦                  ¦
ux uy uz *                 ¦------X           * ux uy uz
        ¦                                     ¦
        ¦                                     ¦
        ¦                                     ¦
     θy *                                     * θy
        ¦                                     ¦
        ¦                                     ¦
        ¦                                     ¦
        * ------ * ------- * ------- * ------ *
ux uy uz        θx    ux uy uz      θx       ux uy uz
```

Figure 1.6   The SemiLoof nodal degree of freedom set. (32 DOF)

```
ux uy uz - - -        ux uy uz θx θx -        ux uy uz - - -
               * --------------- * --------------- *
               ¦                                   ¦
               ¦                                   ¦
               ¦                                   ¦
               ¦                 Y                 ¦
               ¦                 ¦                 ¦
               ¦                 ¦                 ¦
               ¦                 ¦                 ¦
     ux uy uz  *                 ¦-----X           * ux uy uz
     θy θy -   ¦                                   ¦ θy θy -
               ¦                                   ¦
               ¦                                   ¦
               ¦                                   ¦
               ¦                                   ¦
               ¦                                   ¦
               ¦                                   ¦
               * --------------- * --------------- *
ux uy uz - - -       ux uy uz θx θx -         ux uy uz - - -
```

Figure 1.7 Plan view of the ANSYS element host with the Semiloof degree of freedom set accomodated.

```
11 ------12------  13     1...8  Internal element
 ¦ 7 L6   6 L5 5 ¦                 node numbers.
 ¦ L7          L4 ¦
9 ¦ 8    EL 2    4 ¦ 10
 ¦ L8          L3 ¦       L1..L8    Loof nodes.
 ¦ 1 L1   2 L2 3 ¦
6  -------7------  8
 ¦ 7 L6   6 L5 5 ¦        1...13  Node numbers of
 ¦ L7          L4 ¦                the model.
4 ¦ 8    EL 1      ¦ 5
 ¦ L8          L3 ¦
 ¦ 1 L1 2   L2 3 ¦
1  ------2-------  3
```

Figure 1.8 Rotational degree of freedom incompatibility at midside nodes between adjacent elements.

Therefore, in PREP7 after the element type 100 has been selected, the spreadsheet defined in USEREL is read and an eight noded, six degree of freedom per node, element type spreadsheet is provided. Any data subsequently provided for the element will be labelled by PREP7 as x, y, z, θx, θy, θz per node.

During matrix formulation, the element nodal degree of freedom data is passed into ST100. It is then sent to the SemiLoof stiffness matrix subroutines via a set of interfacing subroutines which resolve the degree of freedom set defined by PREP7 into that expected by SemiLoof.

The user's involvement with the nodal degree of freedoms within PREP7 will be to define nodal loads and displacement constraints. It is essential that the user is aware that the degree of freedom headings given by ANSYS will only be correct for the translational degrees of freedom. At the corner nodes, no rotational quantities should be defined, whilst at the midside nodes, only the first two rotations should be defined, and though they are referenced by PREP7 as x and y rotations respectively, they are in fact interpreted by SemiLoof as both being about the same axis. Any rotational quantities defined at the midside nodes will ultimately be calculated for at the positions of the Loof nodes. (See Fig.1.6).

It is not possible for the user to alter the nodal labels given by ANSYS in PREP7.

A further consequence of placing the Loof rotations at the midside nodes is that they can be incorrectly matched with the Loof rotations of adjacent elements.

For example, if the rotations are appropriated to the midside nodes on a first come first served basis (traversing in a consistent direction around the element nodes), then confusion of rotations occur at the boundary with an adjacent element.

Shown in Fig.1.8 (working clockwize around the elements), Loof rotations L1 and L2 from element EL2 are taken by the midside node 7, as are Loof rotations L5 and L6 from element EL1.

This means that at the shared midside node on the boundary between elements 1 and 2, the Loof nodes (L1, L5) and (L2, L6) are taken as being matching degrees of freedom when in fact they are not.

In order to overcome this problem, Loof rotations are appropopublic to the midside nodes depending on the numerical values of the corner node numbers. i.e. If N1 > N2 (N1 and N2 being elemental corner nodes defining a side), then L1 occupies the first position in the midside rotations array locations. If N1 < N2, then L2 will occupy the first position.

Using this logic, the degree of freedom mismatch at midside nodes is overcome.

1.4.2. The order of element nodal definition.

When defining nodes to PREP7 for an eight noded shell, the order of element nodal definition must be as shown in Fig.1.9.

If the element nodal definition is given to PREP7 as anything except corner - corner - corner - corner - midside - midside-midside - midside, the element will effectively have been defined in PREP7 as being twisted and warped.

Once the element has been correctly defined (nodally), PREP7 writes the nodes, in order of definition, to FILE27 for input to the solution phase.

```
ux uy uz θx θy θz      ux uy uz θx θy θz        ux uy uz θx θy θz
                  * --------------- * --------------- *
                  | 4               7               3 |
                  |                                   |
                  |                                   |
                  |                 Y                 |
                  |                 |                 |
                  |                 |                 |
                  |                 |                 |
      ux uy uz * 8                 |-----X          6 * ux uy uz
      θx θy θz |                                      | θx θy θz
                  |                                   |
                  |                                   |
                  |                                   |
                  |                                   |
                  |                                   |
                  | 1               5               2 |
                  * --------------- * --------------- *
ux uy uz θx θy θz      ux uy uz θx θy θz        ux uy uz θx θy θz
```

Figure 1.9 Order of elemental nodal definition expected by PREP7 for eight noded shells.

```
ux uy uz - - -         ux uy uz θx θx -            ux uy uz - - -
                  * --------------- * --------------- *
                  | 7               6               5 |
                  |                                   |
                  |                                   |
                  |                 Y                 |
                  |                 |                 |
                  |                 |                 |
                  |                 |                 |
      ux uy uz * 8                 |-----X          4 * ux uy uz
      θy θy -  |                                      | θy θy -
                  |                                   |
                  |                                   |
                  |                                   |
                  |                                   |
                  |                                   |
                  | 1               2               3 |
                  * --------------- * --------------- *
ux uy uz - - -         ux uy uz θx θx -            ux uy uz - - -
```

Figure 1.10 Order of elemental nodal definition expected by SemiLoof

In the solution phase, FILE27 is read and the nodal coordinates are used with their corresponding shape functions and differentiated during the calculation of the Jacobian inverse and determinant. However, when using the SemiLoof shell, the shape function polynomial terms are ordered differently to those of ANSYS eight noded elements and the nodal coordinates must therefore be reordered before being passed to SemiLoof.

SemiLoof expects the nodes to be defined as shown in Fig.1.10

The reordering of the nodes and hence coordinates, is done in subroutine INTERIN (Sect.1.4.4), before the coordinates are used in the shape function derivitives required to calculate the Jacobian inverse and its determinant.

Once the SemiLoof elemental stiffness matrix has been assembled, its rows and columns, and hence its nodal degree of freedom arrangement, are representitive of Fig.1.10.

This means that the rows and columns must be swapped back to respect the nodal degree of freedom order of ANSYS as shown in Fig.1.9, before being submitted for solution so as to correspond correctly with the items in the ANSYS nodal force vector. The 32x32 SemiLoof stiffness matrix must also be expanded to a standard ANSYS 48x48 stiffness matrix by inserting null rows and columns in the appropriate places. Average values of matrix stiffness are placed in the leading diagonal at the junctions of all null rows and columns to effect mathematical stability.

The process of swapping rows and columns, together with the necessary matrix expansion is carried out in subroutine INTEROUT, (Sect.1.4.4).

1.4.3. Programming Inconsistencies

Because ANSYS and SemiLoof had two completely different origins, the programming style is different. Certain Semiloof variable and array name meanings contradict those used by ANSYS, whilst others are not recognised at all.

Arrays for data storage are arranged differently, a problem magnified by the different sizes of array used by each program. (e.g Size and arrangement of the nodal coordinates arrays, stiffness matrix, etc.).

Another inconsistency is that ANSYS always uses Double Precision dimensioning wheras the original SemiLoof subroutines use Single Precision. This has been changed in order to avoid the incorrect reading of data.

All ANSYS variable and array names are redefined in SemiLoof terminology in subroutine INTERIN before being passed to the SemiLoof subroutines.

1.4.4 Interfacing the Subroutines

In summary, the inconsistencies to be overcome were:

a. Incompatible nodal degrees of freedom set used by ANSYS and Semiloof.

b. Order of elemental nodal definitions (shape function polynomial terms).

c. Incompatible variable and array names. Different variable storage sizes.

The above incompatibilities were overcome by introducing interfacing subroutines which served to modify the element data between ANSYS and SemiLoof. The subroutines front and back ended the SemiLoof subroutines. All variables and arrays within the SemiLoof subroutines were redimensioned using Double Precision storage.

The complete set of subroutines used in the interfacing of SemiLoof with ANSYS are listed in Table 1.6. Their calling order is shown in Table 1.7 and explained below:

1.4.4.1. ANSYS calls ST100.

Element geometry, (i.e. Nodal coordinates, nodal element definitions), material properties (e.g. Young's modulus, Poisson's ratio), element pressures and thicknesses are passed to ST100 via the ST100 calling arguments.

1.4.4.2. ST100 calls INTERIN.

INTERIN is the interfacing subroutine which converts ANSYS data into data suitable for SemiLoof.

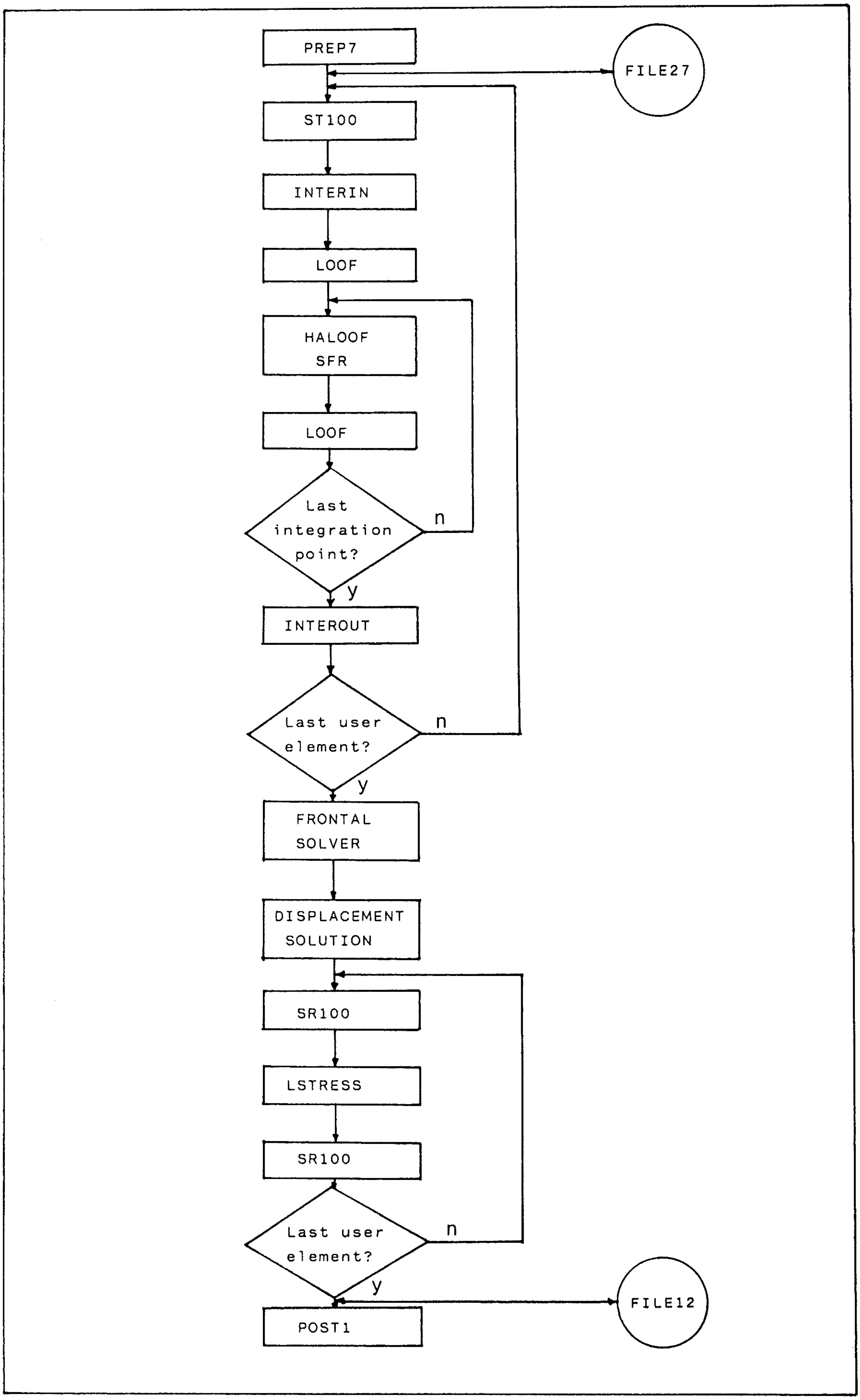
INTERIN performs the following functions:

a. ANSYS variable and array names are equivalenced or converted to SemiLoof specification.
b. Nodal coordinates from ANSYS are assigned to Semiloof nodes (Figs.1.9,1.10).
c. Material properties, element thicknesses and nodal coordinates are taken from ST100 and given new names. Element pressures are passed to INTERIN.

Table 1.6 Complete set of user defined subroutines used in the implementation of the SemiLoof shell into ANSYS.

USEREL	- ANSYS origin. Used by PREP7 to check data input.
USERPT	- ANSYS origin. Used by PREP7 to formulate element plots.
ST100	- ANSYS origin. Used at the matrix formulation stage to access user stiffness matrix generation routines.
INTERIN	- User Programmer origin. Converts ANSYS stiffness matrix data to SemiLoof specification.
LOOF	- SemiLoof origin. Modified to recieve data from INTERIN. Drives remaining SemiLoof subroutines.
HALOOF	¦
SFR	¦
BLOCK DATA	¦ - SemiLoof origin. Creates SemiLoof stiffness
VECTOR	¦ matrix.
SCALAR	¦
INTEROUT	- User Programmer origin. Converts SemiLoof stiffness matrix into ANSYS format.
SR100	- ANSYS origin. Extensive modifications. Calculates stresses and writes them to the ANSYS post data FILE12.

Table 1.7 Data flow diagram showing subroutines and files used during the processing of a SemiLoof shell.



1.4.4.3. INTERIN calls LOOF

LOOF is the doorway to the SemiLoof routines. LOOF reads the reordered nodal coordinates, material properties and element thicknesses under the new names given to them by INTERIN.

LOOF dimensions and initialises all arrays and variables to be used by SemiLoof.

LOOF is mainly original SemiLoof.

1.4.4.4. LOOF calls the SemiLoof subroutines in an integration point loop (i.e. four times per element).

The SemiLoof subroutines kept in their original form were:

HALOOF, SFR, SCALAR and VECTOR.

LOOF calls HALOOF which then calls SFR, SCALAR and VECTOR as required.

The Semiloof subroutines calculate the required data:

i.e. Integrated shape function derivitives. Constituents of the [B] matrix.

These items are then passed back to LOOF on an integration point basis. i.e. HALOOF is called by LOOF four times per element stiffness matrix calculation.

Each return to LOOF delivers the terms of an integrated [B] matrix for each integration point in the element. The elasticity matrix [D] is then calculated and the multiplication of $[B]^T[D][B]$ commences. The resulting numerical values are then sent to their appropriate locations in the 32x32 SemiLoof stiffness matrix.

Finally, when on completion of the final integration point data multiplication, and the elemental 32x32 SemiLoof stiffness matrix has been fully assembled, it is sent back to ANSYS via another interfacing subroutine INTEROUT, which converts the 32x32 SemiLoof matrix to a 48x48 ANSYS compatible matrix.

Another function performed by LOOF is to carry out the [D][B] multiplication required for calculating stresses. Each elemental [D][B] matrix is written to FILE03 via a set of saved variables (SVR).

They are passed to FILE03 via a call to PUTELD (ANSYS origin).

Also calculated within LOOF are the equivalent nodal pressure loads which are saved to a 32x1 column array (ZASS).

1.4.4.5. LOOF calls INTEROUT

INTEROUT reads the SemiLoof 32x32 stiffness matrix, and performs three operations:

a. The 32x32 SemiLoof stiffness matrix is expanded to fit a 48x48 matrix as expected by ANSYS. Zeroes are inserted in the superflous rows and columns.

b. The rows and columns of the 48x48 matrix are re-ordered to complement the ANSYS element nodal definitions specified in Fig.1.9.

c. The 32x1 SemiLoof equivalent nodal pressure vector is reordered and expanded to a 48x1 ANSYS format equivalent nodal pressure vector to be submitted along with the 48x48 stiffness matrix for displacement solution.

1.4.4.6. INTEROUT returns to INTERIN and then back to ST100

The items passed back to ST100 are the 48x48 ANSYS nodally ordered stiffness matrix and a similarly ordered 48x1 nodal force vector. These items are passed in arrays ZS and ZASS respectively.

1.4.4.7. ST100 returns to the MAIN program

ST100 returns to the main program and stores the element matrices for later wave front solution. ANSYS then moves on to the next element in the list and if it is a user element (Stif100), ST100 is again called and the process detailed in Sections 1.4.4.1 to 1.4.4.7 is repeated.

After all element matrices have been formed, ANSYS procedes with the displacement solution and passes the displacements to SR100 for stress calculations.

1.4.4.8. MAIN program calls SR100

SR100 is the subroutine which is used to calculate the element stresses and other post displacement items. The element nodal displacements are read by SR100. Subroutine LSTRESS (User Programmer origin), is then called which calculates the elemental stresses at each integration point and passes them back to SR100.

1.4.4.9. SR100 calls LSTRESS

LSTRESS takes displacements passed through SR100 in the ANSYS 48x1 vector format. They are then converted to a SemiLoof 32x1 displacement vector format. The [D][B] matrix corresponding to the current integration point is read from FILE03 (using the SVR array), and is post multiplied by the displacements to calculate the component element forces and bending moments.

1.4.4.10. LSTRESS returns to SR100

After return to SR100, labels are assigned to each of the post displacement items recieved from LSTRESS, and are then written to FILE12. The ANSYS postprocessor, POST1, may then used to read FILE12 where the post displacement items are referenced by their item numbers (see Table 1.9).

1.5 Using the SemiLoof shell in ANSYS

1.5.1 Introduction

This Chapter is intended as a User Guide for the SemiLoof shell element in ANSYS. The necessary information is presented in a documentation style similar to that used in the ANSYS User Manual element library, (1).

1.5.2 Stif100: The SemiLoof Shell

1.5.2.1 Input Data

The SemiLoof shell is an eight noded isoparametric quadrilateral structural shell with membrane and bending capabilities.
The geometry, nodal point locations, loading stresses and the coordinate system for this element are shown in Fig.1.11.
The element geometry is defined by eight nodal points and one constant thickness.
The element loading can be either a constant pressure loading, a nodal force loading or an initial nodal displacement.
During solution the pressure loading is resolved into equivalent nodal forces. The element is defined in three dimensional space.
Nodal degrees of freedom vary throughout the element. Every node has three translational degrees of freedom, and each midside node has an additional pair of rotational degrees of freedom.
The rotational degrees of freedom defined at any one midside node rotate about an axis parrallel to the element edge but are calculated away from the midside node at the positions of the two flanking Loof nodes.
A detailed description of the required input data is given in Table 1.8 and illustrated in Fig.1.11.

Table 1.8 SemiLoof shell input data specification

ELEMENT NAME	STIF100	
NO. OF NODES	8	I,J,K,L,M,N,O,P
DEGREES OF FREEDOM PER NODE	3	I,J,K,L (corner nodes) UX,UY,UZ
	5	M,N,O,P (midside nodes) UX,UY,UZ,ROTa,ROTb
REAL CONSTANTS	1	THICKNESS
MATERIAL PROPERTIES	2	EX,NUXY
PRESSURES	1	P1

where EX = Young's modulus.

NUXY = Poisson's ratio.

ROT(n) = Rotation about the element edge.

1.5.2.2 Output Data

Element post displacement data is available for access by the ANSYS postprocessors and is organised as shown in Table 1.9.

1.5.2.3 Element coordinate system and pressure loading

The element coordinate system affects the sign convention of pressure loading and output stresses and moments.

The element x-direction is defined to be towards node J from node I. The element y-direction is perpendicular to x in the element plane and in the general direction of node P from node I.

Using this convention, positive applied pressures act in the z direction (assuming a right handed system). Element stresses and moments follow the convention shown in Fig.1.11.

1.5.2.4 Assumptions and Restrictions

Elements must be nodally defined in a consistant direction from nodes I to P as shown in Fig.1.11. Zero thickness or zero area elements are not allowed.

No corresponding consistent mass or damping matrices are currently available for the SemiLoof shell in ANSYS, therefore any structure mass and damping must be defined at the nodes using ANSYS Stif21 Lumped Mass elements and appropriate discrete viscous dashpot elements.

The SemiLoof shell is only compatible with ANSYS elements having translational degrees of freedom. No current ANSYS elements have a compatible rotational degree of freedom set.

This version of SemiLoof is linear, therefore no plasticity or large rotations can be accounted for.

Table 1.9 SemiLoof shell post displacement data item table.

Centroidally averaged forces

1. TX
2. TY
3. TXY
4. MX
5. MY
6. MXY

Centroidally averaged stresses

ELEMENT TOP PLANE	ELEMENT MIDDLE PLANE	ELEMENT BOTTOM PLANE
7. SXT	10. SXM	13. SXB
8. SYT	11. SXY	14. SYB
9. SXYT	12. SXYM	15. SXYB
16. SG1T	19. SG1M	22. SG1B
17. SG2T	20. SG2M	23. SG2B
18. SG3T	21. SG3M	24. SG3B
25. SGET	26. SGEM	27. SGEB

KEY:
- TX Direct force in element x direction per unit length.
- TY Direct force in element y direction per unit length.
- TXY In plane shearing force per unit length.
- MX Mom. about face normal to element y axis per unit len.
- MY Mom. about face normal to element x axis per unit len.
- MXY Twisting mom. about face normal to x and y axes.
- SX Direct stress in element x direction per unit length.
- SY Direct stress in element y direction per unit length.
- SXY In plane shearing force per unit length.
- SG1, SG2, SG3 Principal stresses SG1 > SG2 > SG3.
- SGE Von Mises Equivalent Stress.

The T,M and B suffixes represent items at the top, middle and bottom surface of the element respectively.
fig.1.11

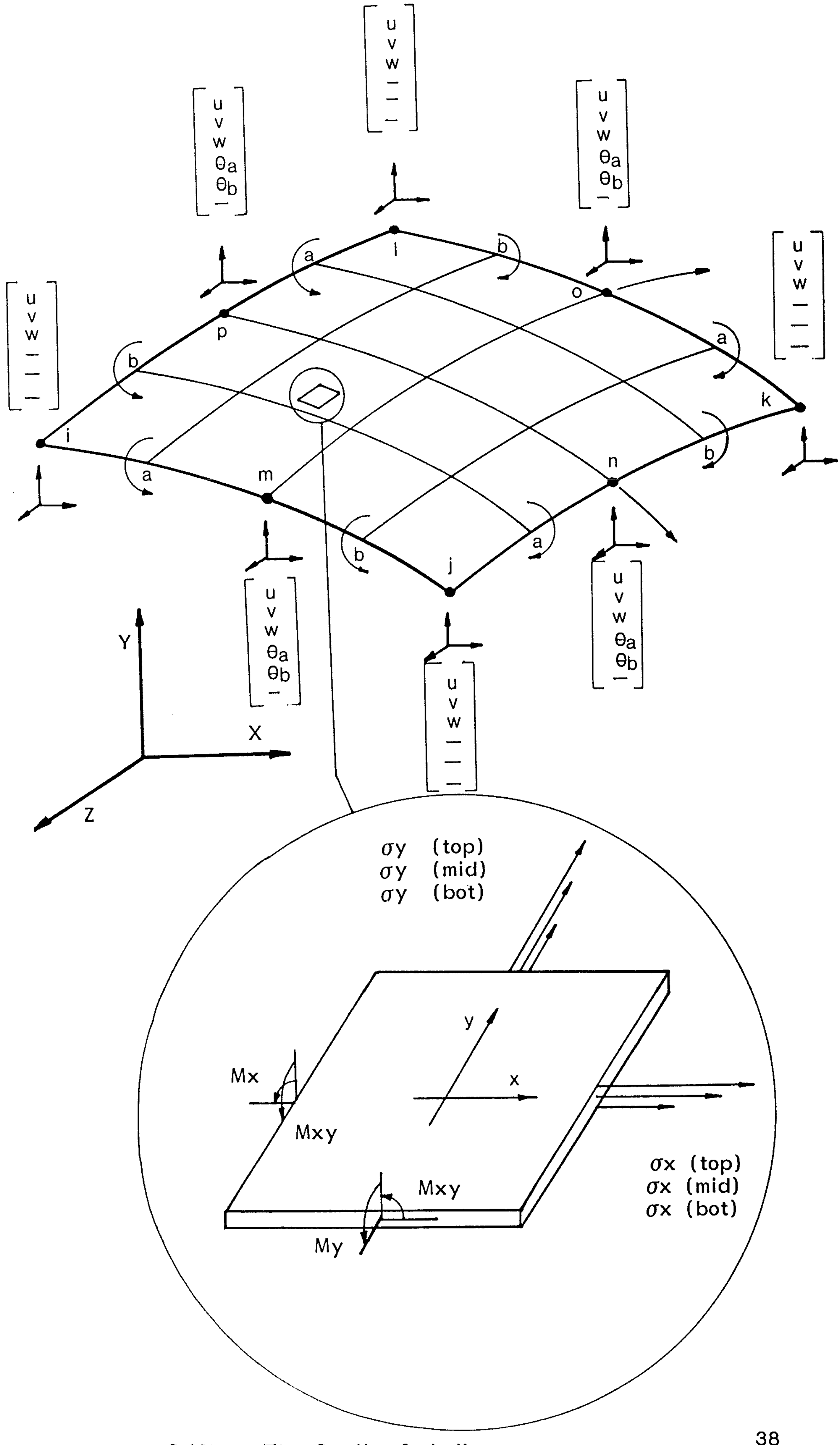


Figure 1.11 Stif100: The SemiLoof shell.
Nodal degrees of freedom and post data items.

1.5.2.5 Stress calculations

The SemiLoof shell element calculates six items at the mid surface (neutral axis):

TX	Direct force in element x direction per unit length.
TY	Direct force in element y direction per unit length.
TXY	In plane shearing force per unit length.
MX	Moment about element x axis per unit length.
MY	Moment about element y axis per unit length.
MXY	Twisting moment normal to the xy plane.

Direct stresses calculated from direct forces have been calculated at the top, middle and bottom surfaces (8):

Dsx = Tx/thickness

Dsy = Ty/thickness

Dsxy = Txy/thickness

where Ds(n) = Direct stresses calculated from direct forces in the element (n) direction at the top, middle and bottom surfaces of the element.

Bending stresses have been calculated from the mid surface moments from the relation (8):

Bsx = Mx * 6 /thickness**2

Bsy = My * 6 /thickness**2

Bsxy = Mxy * 6 /thickness**2

where Bs(n) = Bending stresses at the top and bottom of the element resulting from bending about the mid surface of the element.

Stresses resulting from bending are zero at the element mid surface.

The total stresses at the top, middle and bottom surfaces of the element have been calculated from:

σx (top) = Dsx + Bsx (top)

σy (top) = Dsy + Bsy (top)

σxy (top) = Dsxy + Bsxy (top)

σx (mid) = Dsx

σy (mid) = Dsy

σxy (mid) = Dsxy

σx (bot) = Dsx - Bsx (bot)

σy (bot) = Dsy - Bsy (bot)

σxy (bot) = Dsxy - Bsxy (bot)

1.5.2.6 Principal stresses

Principal stresses have been calculated at the top, middle and bottom surfaces from component stresses averaged at the element centroid (9):

p1 = (σx + σy)/2 + 0.5 (sqrt [((σx -σy)**2) + 4(σxy**2))])

p2 = (σx + σy)/2 - 0.5 (sqrt [((σx -σy)**2) + 4(σxy**2))])

where p1 = Major principal stresses at top, middle and bottom surfaces.

p2 = Minor principal stresses at top, middle and bottom surfaces.

Pl is algebraically the greater of the two principal stresses.

To be compatible with the Stif93 in ANSYS and also to facilitate the calculation of Von Mises equivalent stresses, the two principal stresses are converted to SIG1, SIG2 and SIG3.

where SIG1, SIG2, SIG3 = Principal stresses

and SIG1 > SIG2 > SIG3

SIG1 >= 0

SIG2 >=< 0

SIG3 <= 0

1.5.2.7 Von Mises Equivalent stresses

Von Mises stresses are calculated from (1):

$$SIGE = 1/SQRT(2)*SQRT\ [(SIG1-SIG2)^2 + (SIG2-SIG3)^2 + (SIG3-SIG1)^2]$$

where SIGE = Von Mises equivalent stresses

CHAPTER 2

FINITE ELEMENT MODELLING OF A REINFORCED CONCRETE NUCLEAR SHELTER

2.1 Introduction

The shelter investigated in this thesis is detailed in the Home Office publication 'Domestic Nuclear Shelters - Technical Guidance' (11). It is a domestic, six person, shallow buried nuclear shelter constructed of reinforced concrete as shown in Fig.2.2 and Appendix E. An elementary finite element analysis has already been conducted on the shelter by a Home Office working party whose results have been the basis for the design of a shelter published in (11).

As the shelter is designed as a protective structure, any damage that is incurred by the blast load must not compromise the operation of the shelter. It is intended that the occupants should inhabit the compartment sealed by the blast door. The stairwell area is intended for access only. It must be borne in mind that the single compartment shelter may have to house six (and possibly more) people in the one room for up to three weeks after the damage from the blast loading has occured. In these cramped conditions, a comfortable environment is of prime importance for the survival and maintenance of health of the occupants. Cracked and leaking walls, or inoperative ventilation and drainage systems cannot be tolerated if the occupants are to eventually emerge to an above ground environment which may offer few facilities for medication and recovery.

The most important structural consideration is the maintenance of the escape routes. The access hatch is situated above ground and is therefore subjected to additional loading of wind, flying debris and collapsing buildings. The hatch may be permitted to undergo slight damage and it is expected that the occupants of the shelter will be able to clear an exit if necessary. Of prime importance is the preservation of the blast door, which is mounted in the 400mm thick blast wall. Apart from preventing the failure of the door itself, any permanent or excessive elastic deformation of the mounting wall could cause the door to jam, thus rendering escape from the shelter impossible. This must be avoided at all costs.

The approach taken by the Home office working party was to base their design on a structure allowed to undergo moderate permanent damage, i.e. "there will be considerable yielding of steel and cracking of concrete" (11), though in view of the above points, it is proposed in this thesis that an appropriate shelter design should be one which allows the shelter to remain undamaged. It is important to ascertain whether the ultimate design loads would in fact be reached in response to the specified blast load, as it may be found that the actual bending moments are less severe than originally calculated and that the shelter is overdesigned; thus although the shelter was originally designed to succumb to moderate damage, it may yet serve the suggested criteria of little or no damage. Should the results of the re-analysis not correlate at all with those of the Home Office working party, it may be recommended that a full nonlinear analysis be conducted in order to establish realistic design data.

In this thesis, the Home Office analyses (linear static and dynamic), were re-run using SemiLoof shell elements in ANSYS. This chapter reviews the original design criteria, and then states the assumptions and simplifications made in the finite element model.

2.2 Home Office shelter design philosophy

The shelter is constructed of doubly reinforced concrete. All walls are 250mm thick except the wall containing the blast door which is 400mm thick. The blast door is constructed of 15mm thick steel. The shelter is 'shallow buried' (i.e. the roof is 300mm below ground), with the stairwell access hatch protruding above the ground surface to provide access to and from the shelter. The structural detailing for the shelter is shown in Appendix E (note the reinforcement used in this thesis is slightly different - Table 1.2). The Home Office publication "Domestic Nuclear Shelters - Technical guidance" (11), sets out design guidelines in accordance with the now superseded CP110. The blast resistant design process documented in (11), uses the following methodolgy:

a. The shelter is designed to withstand moderate permanent damage, on the basis of an elastic response of a reinforced concrete structure, subject to a factored load. The factored load is calculated from the equation:

$$ru = F(1/(1-(1/2u))$$

where ru = ultimate unit resistance

F = Blast load

u = Ductility ratio

For moderate damage, u=3, therefore ru = 1.2F.

Hence the shelter will be designed to withstand a load 1.2 times greater than that anticipated, whilst undergoing moderate permanant damage.

b. The strength of the materials can be increased due to their ductility, by the following dynamic increase factors:

Stresses	Dynamic increase factor
Steel - bending	1.10
Concrete - compression	1.25

Hence Fy(dynamic) = 1.10(Fy)

Fcu(dynamic) = 1.25(Fcu)

where Fy = Characterstic strength of steel reinforcement.

Fcu = Characteristic concrete cube strength.

c. The reinforced concrete panels were designed by the ultimate moment philosophy:

Mu(steel) = fy(dynamic).As.z

Mu(conc.) = 0.225fcu(dynamic).b.d^2

where Mu = Ultimate moment of resistance

z = lever arm of internal moment of resistance

As = Area of tensile reinforcement

b = Breadth of section

d = Effective depth of section

and z = (1 - (0.84.fy(dynamic).As)/(fcu(dynamic).b.d))d

where z <= 0.95.d

The design for a typical panel is reproduced in Table 2.1 and shows that for a 250mm thick section with R16 reinforcement at 200mm centres, the ultimate moments of resistance are:

Mu(steel) = 0.225(37.5)182**2 = 280 kNm/m

Mu(conc.) = 275(1.005)182 = 51 kNm/m

For the 400mm blast wall, the ultimate moments of resistance are:

Mu(steel) = 0.225(37.5)324**2 = 886 kNm/m

Mu(conc.) = 275(1.005)324 = 90 kNm/m

Table 2.1 Design of a typical concrete panel as recommended in the Home Office guide.

DATA	CALCULATION	RESULTS
	Given: 2440 mm clear span, one way spaning with fixed supports. 250 mm thick slab, with 50 mm cover to steel.	
CP110 Tables 2 and 3	Grade 30 concrete fcu = 30 N/mm^2 Mild steel fy = 250 N/mm^2 Consider 1 mm width of slab.	fcu = 30 N/mm^2 fy = 250 N/mm^2
DIF's from Sect.2.2	STEP 1: DESIGN STRESSES fcu(dynamic) = 1.25 x 30 = 37.5 N/mm^2 fy(dynamic) = 1.1 x 250 = 275 N/mm^2	fcu(dynamic) = 37.5 N/mm^2 fy(dynamic) = 275 $N/mm.^2$
Fig.3.2	STEP 2: BLAST LOAD P = 1 atmos. = 0.103 N/mm^2 (15 psi)	P = 0.103 N/mm^2
For u=3	STEP 3: REQUIRED ru ru for blast load =1.2P = 1.2 x 0.103 = 0.12 N/mm^2 Add dead load conc. + soil = 0.014 -------- 0.134 N/mm^2	Req'd ru = 0.134 N/mm^2
	STEP 4: REQUIRED Mu A fixed one-way spanning slab Mu req'd = ru L2/16 = .134 x 2440^2 /16 = 49861 Nmm/mm	Mu req'd = 49861 Nmm/mm
250 mm slab As = 1.005 mm^2/mm d=250-58 =192	STEP 5: REINFORCEMENT Try R16 @ 200 c/c z = [(1-(0.84Fy(dynamic)As)/(fcu.bd)]d = 192 - (0.84x275x1.005/37.5) = 185.8 or = 0.95x192 = 182 mm	z = 182 mm
	therefore, Mu = fy(dynamic)As.z = 275 x 1.005 x 182 = 50 300 Nmm/mm > 49841 Nmm/mm run	Use R16 @ 200 c/c top and bottom
	STEP 6: CHECK MIN. STEEL Main = .25% x 1 x 192 = 0.48 mm^2/mm = 1.005 mm2/mm provided O.K.	

2.3 Finite element modelling

In this thesis, certain simplifying assumptions regarding the material properties of the reinforced concrete, the description of the foundation supports, the specification of the finite element mesh and the lumping of the structural mass at discrete nodal points have been made. These simplifying assumptions are stated in Sections 2.3.1 to 2.3.3.

2.3.1 Material properties

Reinforced concrete is used throughout the shelter, with the exception of the blast door which is steel.

The SemiLoof shell elements which have been used to model the reinforced concrete have no special calculation features in their current implementation to take account of the nonlinear behaviour of reinforced concrete. i.e. They have no facilities for calculating cracking or crushing of the concrete or the plasticity of the reinforcing bars.

The material properties used for reinforced concrete and mild steel in this thesis are as follows:

	Reinforced concrete	Steel blast door
Young's modulus	28×10^9 N/m^2	210×10^9 N/m^2
Poisson's ratio	0.17	0.30
Density	2552 kg/m^3	7850 kg/m^3

The values of Young's modulus and Poisson's ratio are typical for mild steel, whilst those taken for the reinforced concrete are those of unreinforced concrete taken from CP110.

Typical values of Poisson's ratio for concrete are within the range of 0.15 to 0.2. An average value of 0.17 was used. The concept of an equivalent Poisson's ratio for reinforced concrete is inappropriate because the presence of the reinforcing steel within the concrete simply acts as a grillage and does not contribute to any lateral strains within the concrete.

In order to establish whether it would be valid to factor Young's modulus to take account of the reinforcing steel within the concrete, the following investigation was conducted:

a. A finite element model of a concrete slab complete with reinforcing bars was constructed, where the concrete was represented by brick elements and the reinforcement by beam elements.

The first load case applied to the slab caused a membrane tensile response for which the displacement was calculated. By substituting the calculated displacement into a formula for maximum displacement of an axially loaded plate (13), an equivalent value of Young's modulus was obtained. This value was $34\times10^9 N/m^2$ as opposed to the original values of $28\times10^9 N/m^2$ and $210\times10^9 N/m^2$ for the concrete and steel respectively.

A second load case, this time to test the response of the structure in bending, was conducted and the results showed that Young's modulus remained unchanged at $28\times10^9 N/m^2$ (Fig.2.1).

In order to determine whether the shelter would be most severely subjected to direct or bending loads, a further test was conducted on a box type structure, similar to the shelter, which was subjected to a uniform pressure load. The results showed that the stresses due to bending were several orders of magnitude higher than those caused by axial loading. On this basis it was assumed the the shelter would be more susceptible to bending stresses and

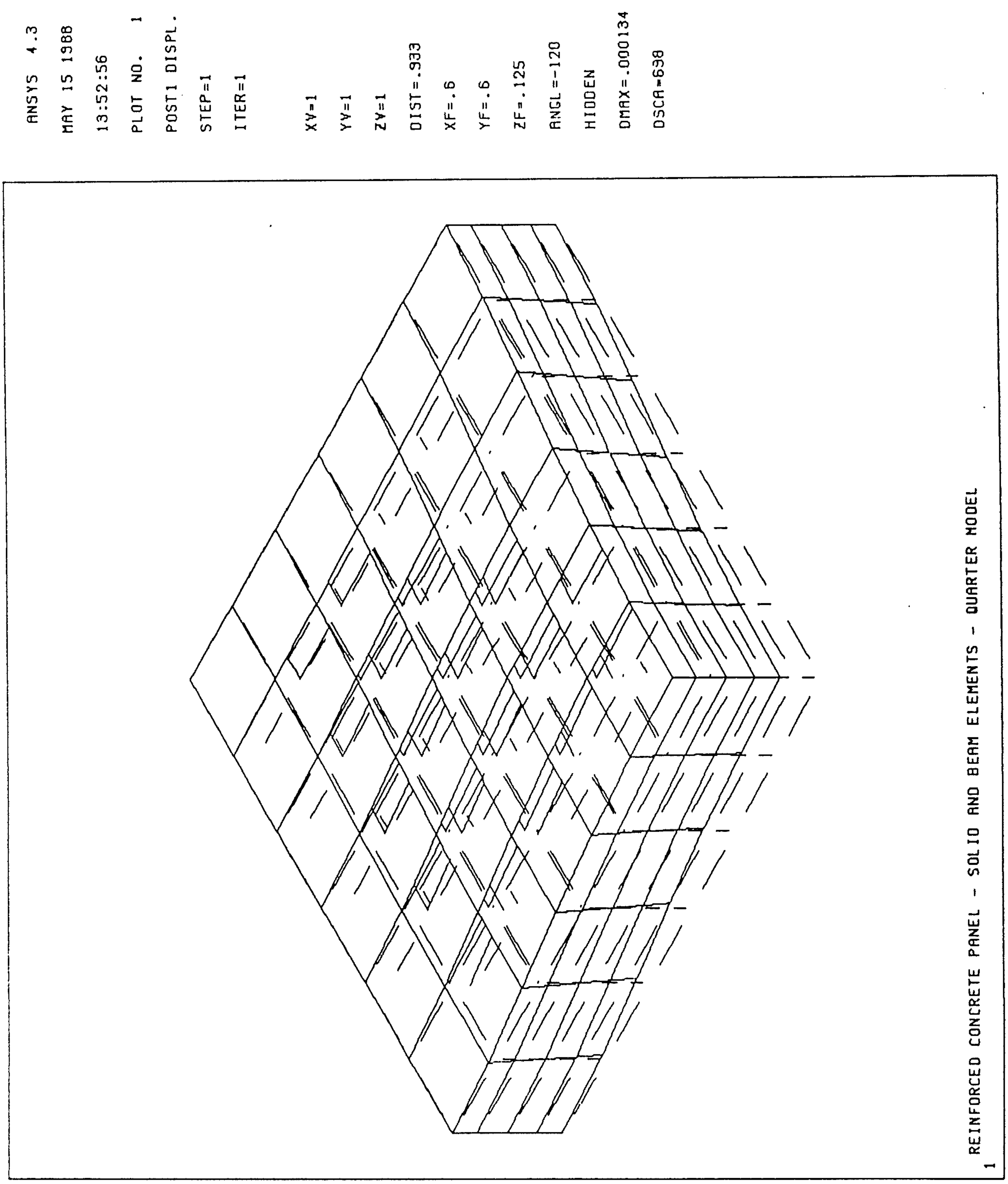


Figure 2.1 Reinforced concrete solid used for material property evaluation.

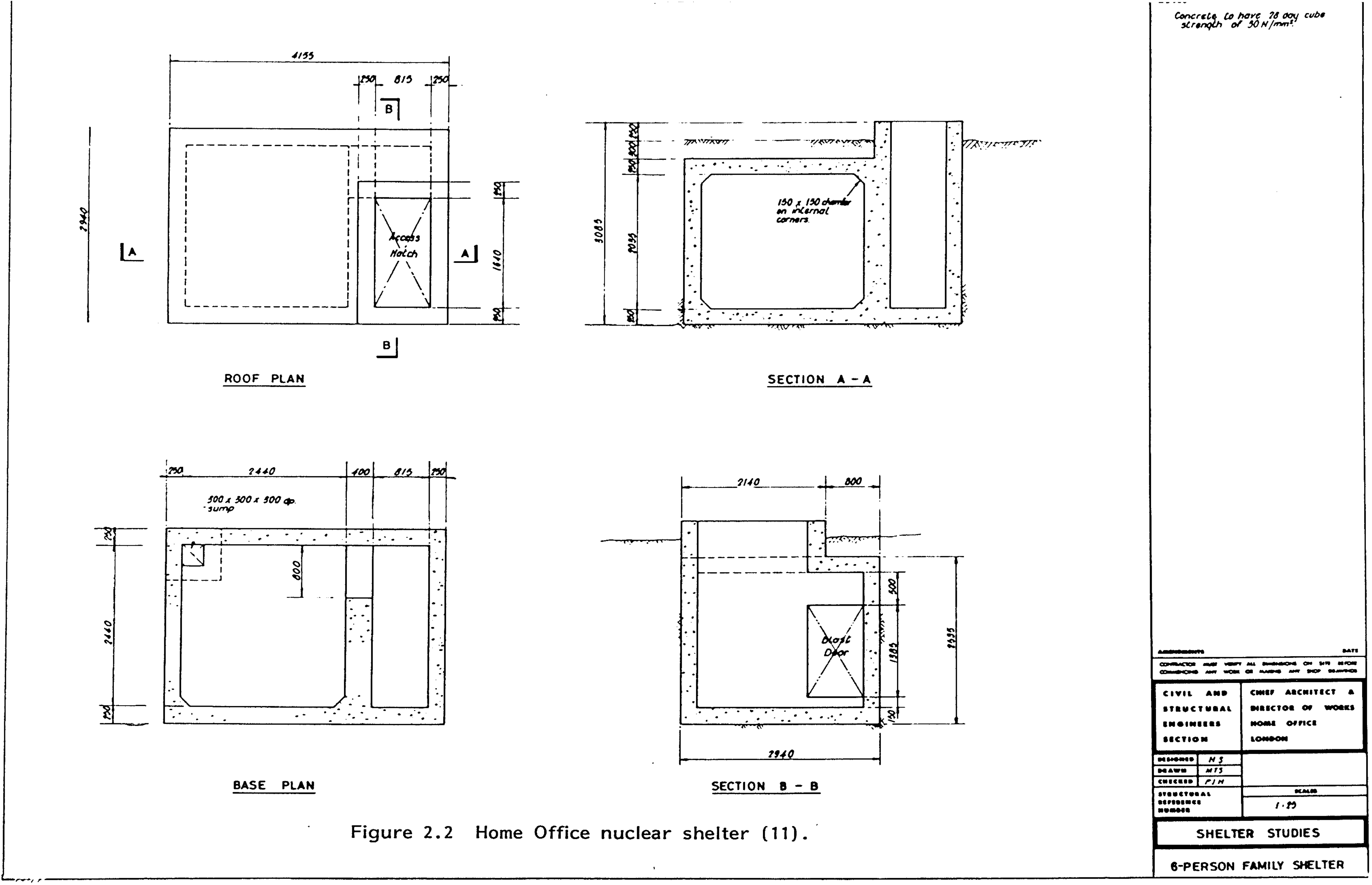


Figure 2.2 Home Office nuclear shelter (11).

therefore a Young's modulus of $28x10^9 N/m^2$ was adopted for this analysis.

A structural analysis based on a material with a lower Young's modulus will characterize lower natural frequencies and higher response amplitudes. The bending moments resulting from the blast load on the more flexible structure will be higher, thus incurring an additional factor of safety in the design.

The equivalent density of the reinforced concrete is calculated from percentage volume of concrete and steel, i.e. with 2.8% reinforcement:

Equivalent density = (2400x0.972) + (7850x.028) = 2552 kg/m^3

2.3.2 Displacement constraints

The displacement constraints adopted by the Home Office working party were of a spring-damper type which simulated an elastic foundation. No details of the soil type or spring stiffness were given in the guide.

In this thesis, the displacement constraints simulated the support conditions of a shelter founded in an excavation in rock, whose sides had not been backfilled. This enabled free expansion and contraction of the shelter as it responded to the blast load (See Fig.2.3). Also, because the foundations were modelled as being rigid, no energy absorption took place within the foundations, resulting in a more severe response of the shelter. The simply supported nature of the shelter also ensured that no local stresses were created due to support conditions.

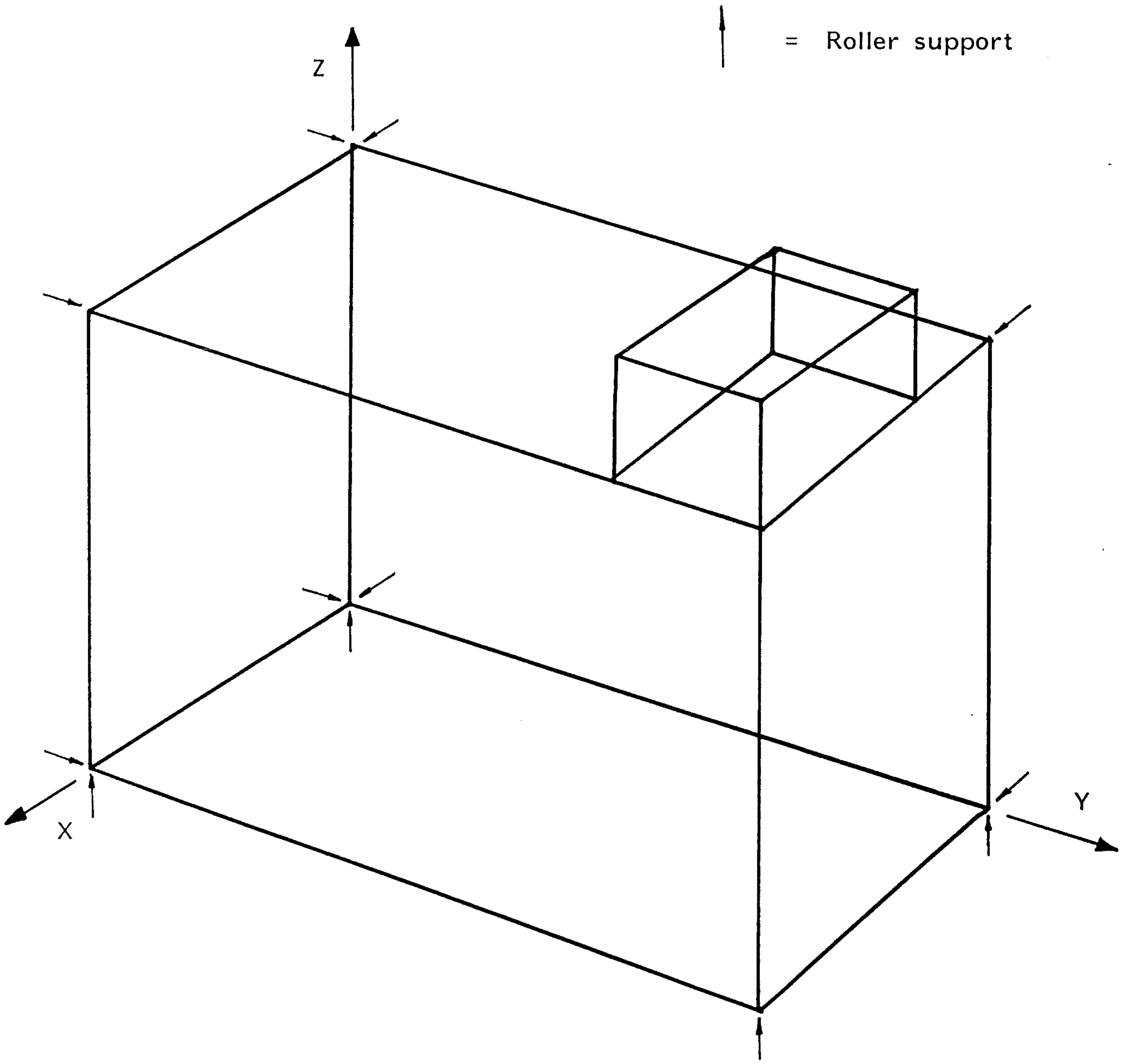


Figure 2.3 Nuclear shelter displacement constraints.

2.3.3 Finite element mesh

The finite element model was made up of SemiLoof shell elements, implemented into ANSYS in Chapter 1. For the dynamic analyses, the structural mass was modelled using ANSYS's lumped mass elements, as no consistent mass matrix was available for the SemiLoof shell. The mesh density and the procedure for the definition of the structural mass will now be discussed.

2.3.3.1 Mesh density

It was decided to construct the finite element model of the shelter with the element arrangement as shown in Figs 2.4 to 2.6.
As the SemiLoof shell is a 'high-order', quadratic displacement element, it was decided that the accuracy gained from a mesh of approximately four elements per wall panel would be sufficient. This assumption is verified by the displacement/mesh convergence data shown in Appendix B - Table B.1, which shows that further gains in accuracy of displacement reduce substantially even after a 2x2 mesh is refined for a fixed plate subjected to a uniform pressure load.
The SemiLoof is a 'thin' shell element, valid for structures having a span/depth ratio of greater than eight, which the shelter has in most of its panels.

2.3.3.2 Lumped mass elements for dynamic analyses

Because the SemiLoof shell implemented in ANSYS has no corresponding consistent mass matrix, the structure mass must be defined by nodal lumping, using discrete ANSYS Stif27 Lumped Mass elements. The distribution of the structural mass to the nodes can significantly affect the dynamic characteristics of the model and

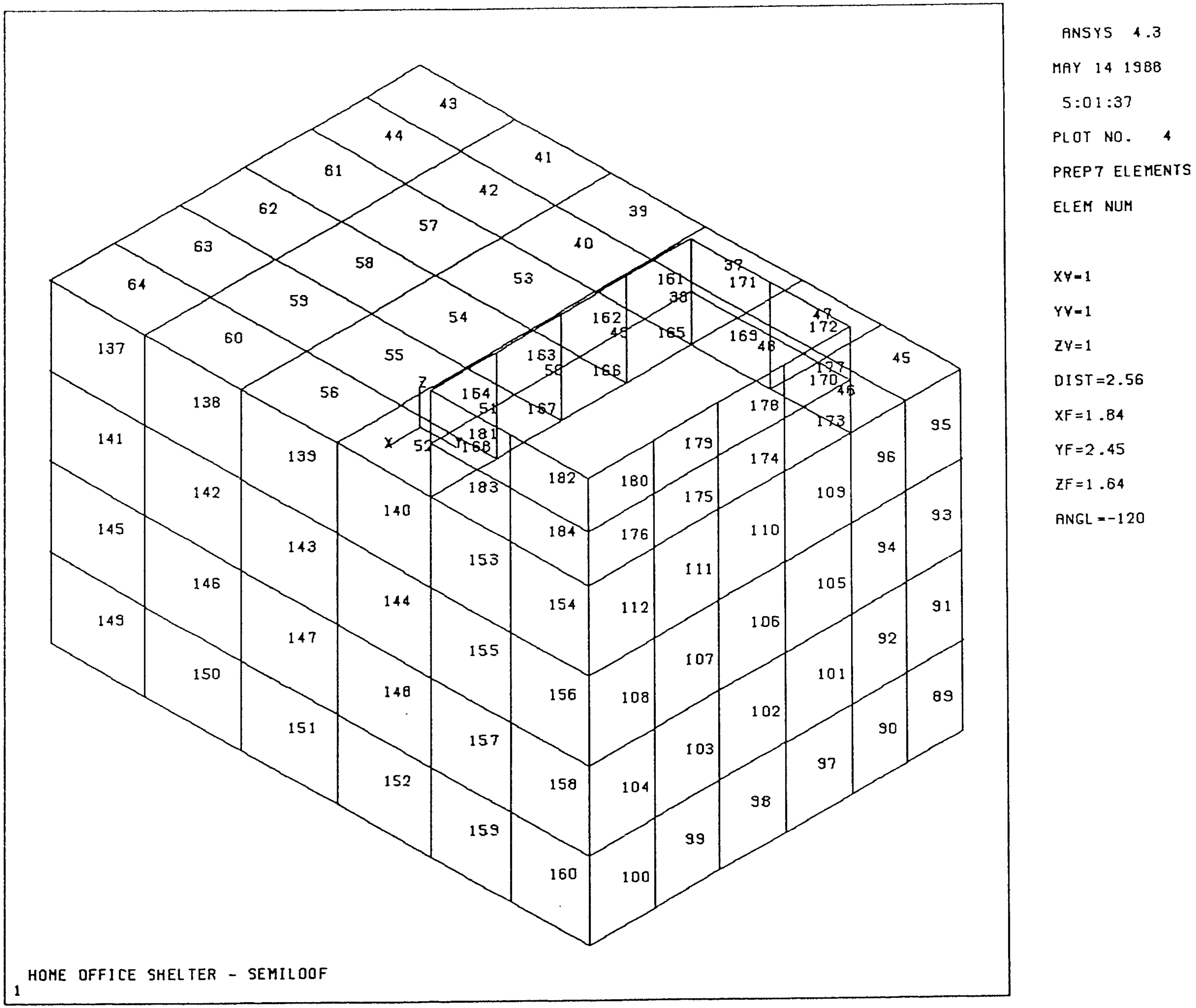


Figure 2.4 Finite element mesh using SemiLoof shell elements.

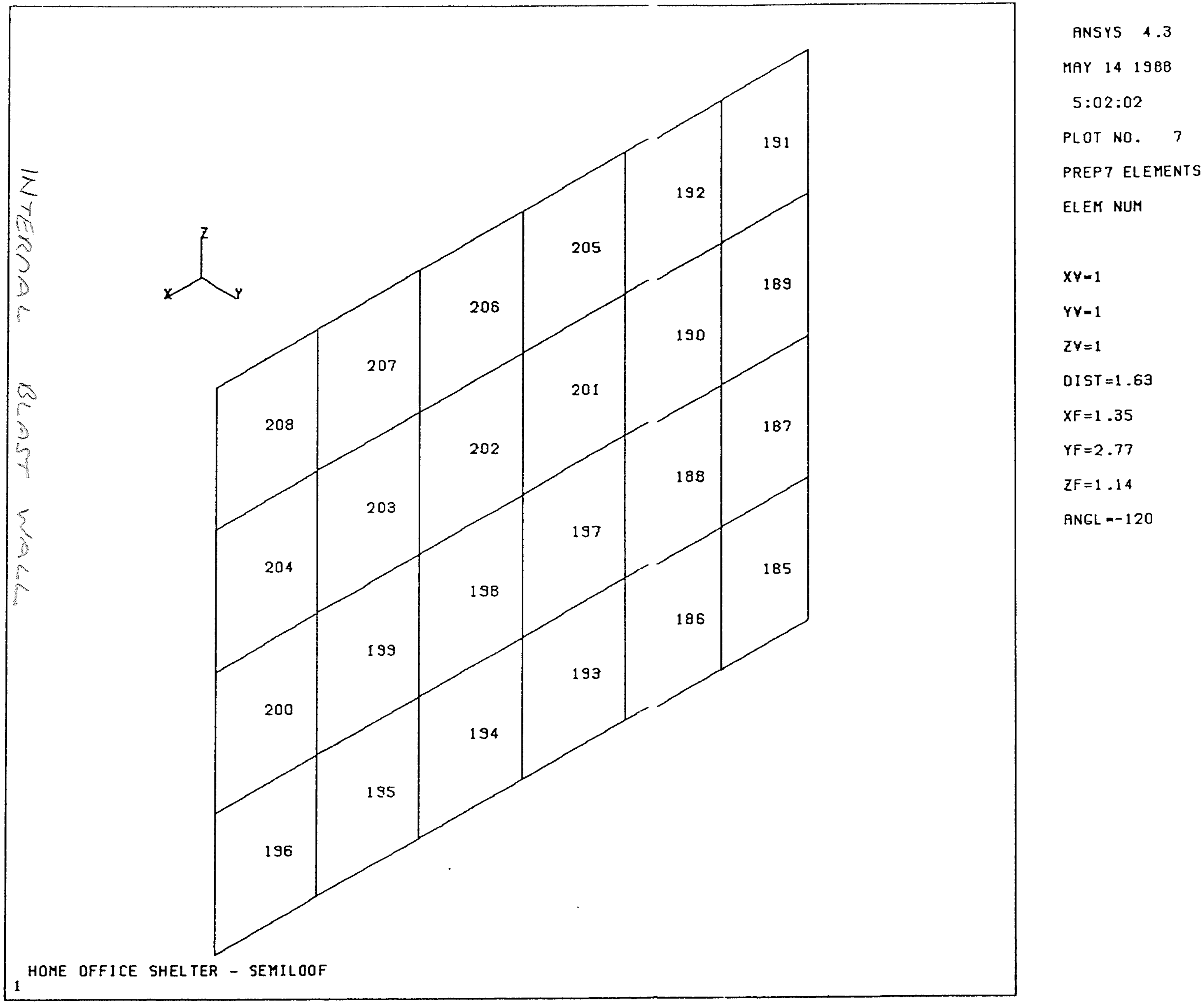


Figure 2.5 Finite element mesh using SemiLoof shell elements.

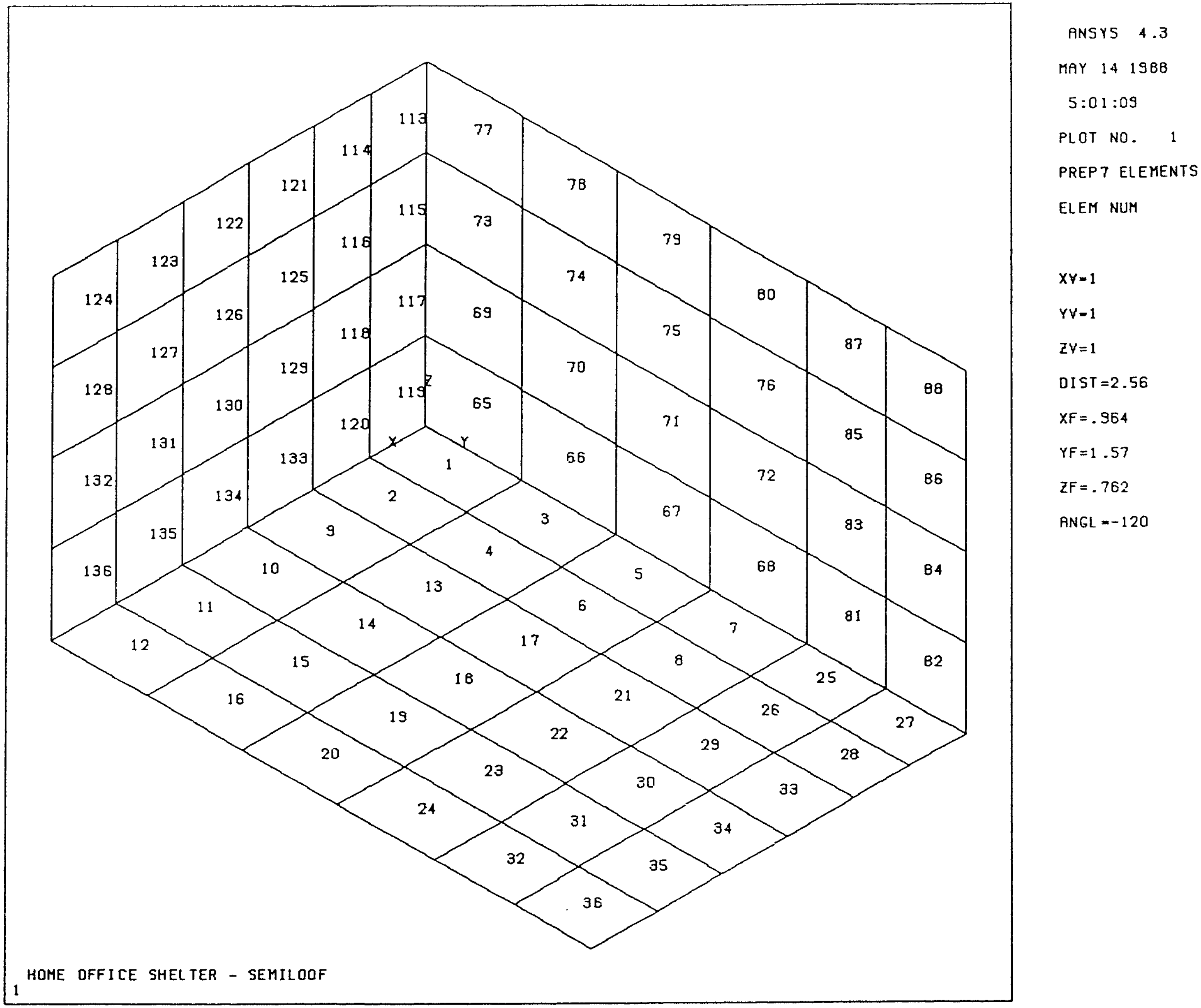


Figure 2.6 Finite element mesh using SemiLoof shell elements.

care must be taken in the nodal mass appropriation.

The technique adopted in this thesis for defining the lumped mass distribution takes advantage of the ANSYS Guyan Reduction (1) capability which can be used to take mass from an element's consistent mass matrix and delegate it to degrees of freedom in the model which exhibit a tendancy in favour of the lower natural frequencies.

The procedure adopted for the structural mass distribution was as follows:

a. The model was first created with ANSYS Stif93 8-node shell elements. The mass was input as density, using Stif93's consistent mass matrix.

b. Fifty dynamic degrees of freedom (lumped mass points) were requested and ANSYS selected the appropriate master degrees of freedom on a high mass/stiffness ratio basis. (Degrees of freedom with high mass/stiffness ratio characterize low frequency, high amplitude modes.)

c. ANSYS then performed a modal analysis, reducing the consistent mass to a lumped nodal mass distribution using Guyan Reduction.

d. During the modal solution, the degrees of freedom chosen by ANSYS for lumped mass distribution were printed into a computer file.

e. This file was then read by the program, MASS.FOR (Appendix C), which took the data and reorganised it into a form suitable for input back into the ANSYS preprocessor. This information specified a node number, a degree of freedom and a quantity of mass for each degree of freedom which had mass distributed to it.

f. The Stif93 model was resumed in PREP7, the Stif93 elements changed to SemiLoof shells and additional Lumped Mass elements

(Stif27's) defined.

g. The input file created by MASS.FOR was then read by PREP7 and mass elements were located at the appropriate nodes, with the corresponding mass acting in the appropriate direction.

The purpose of the methodology in steps a. to g. was to distribute mass within the SemiLoof finite element model on a low frequency basis. The resulting mass distribution is shown in Figs.2.7 to 2.9 and Appendix C.

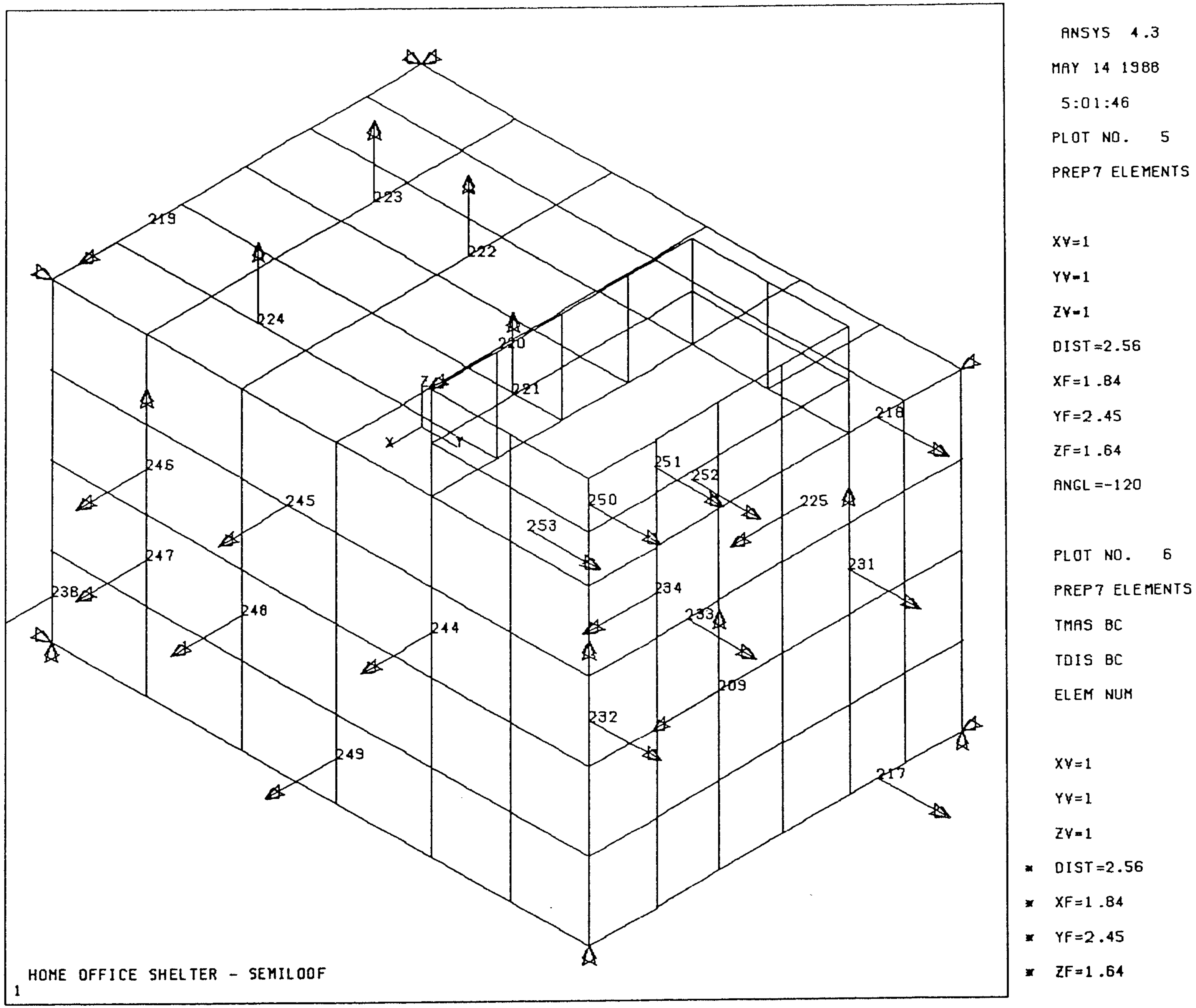


Figure 2.7 Dynamic degrees of freedom.

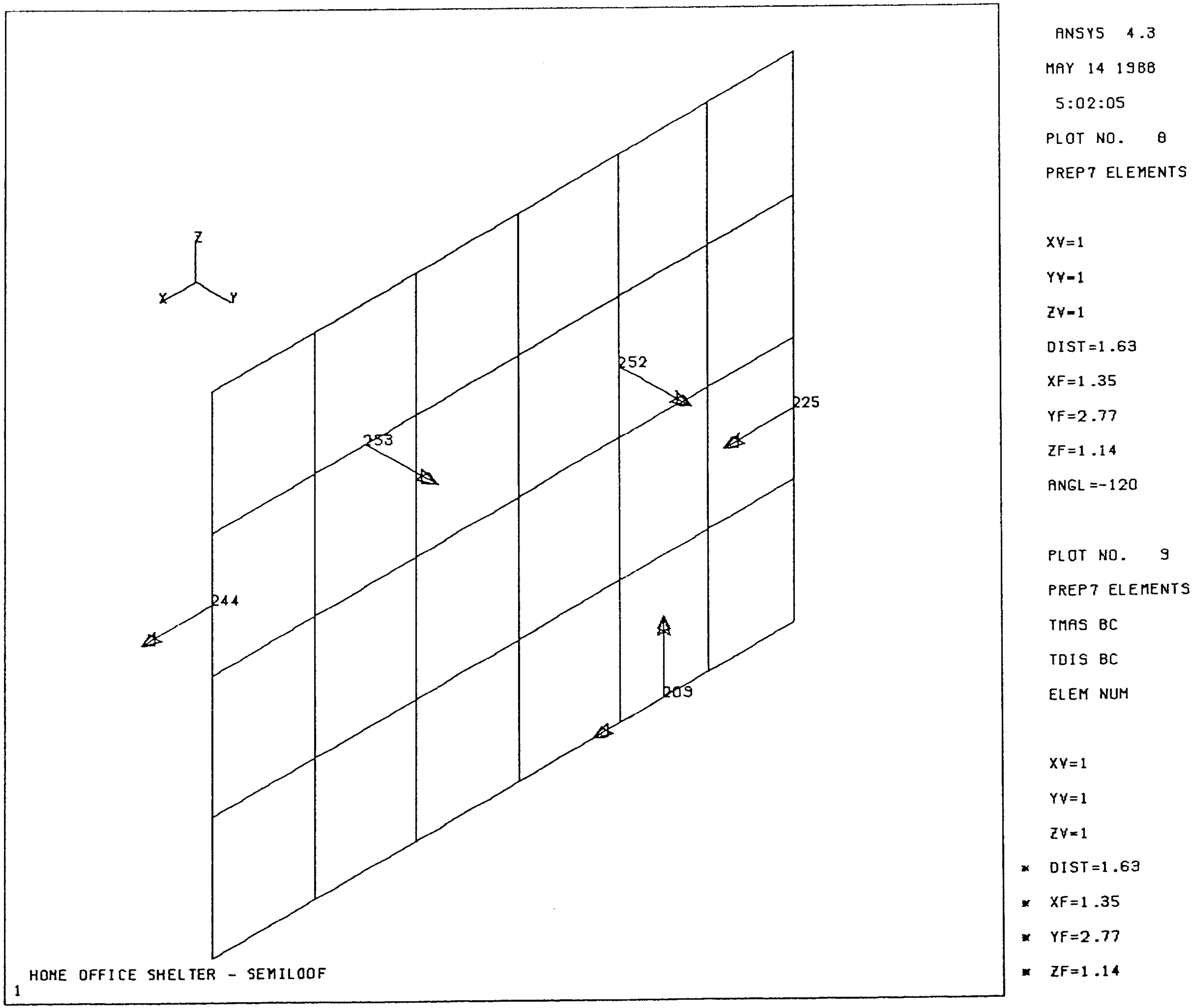


Figure 2.8 Dynamic degrees of freedom.

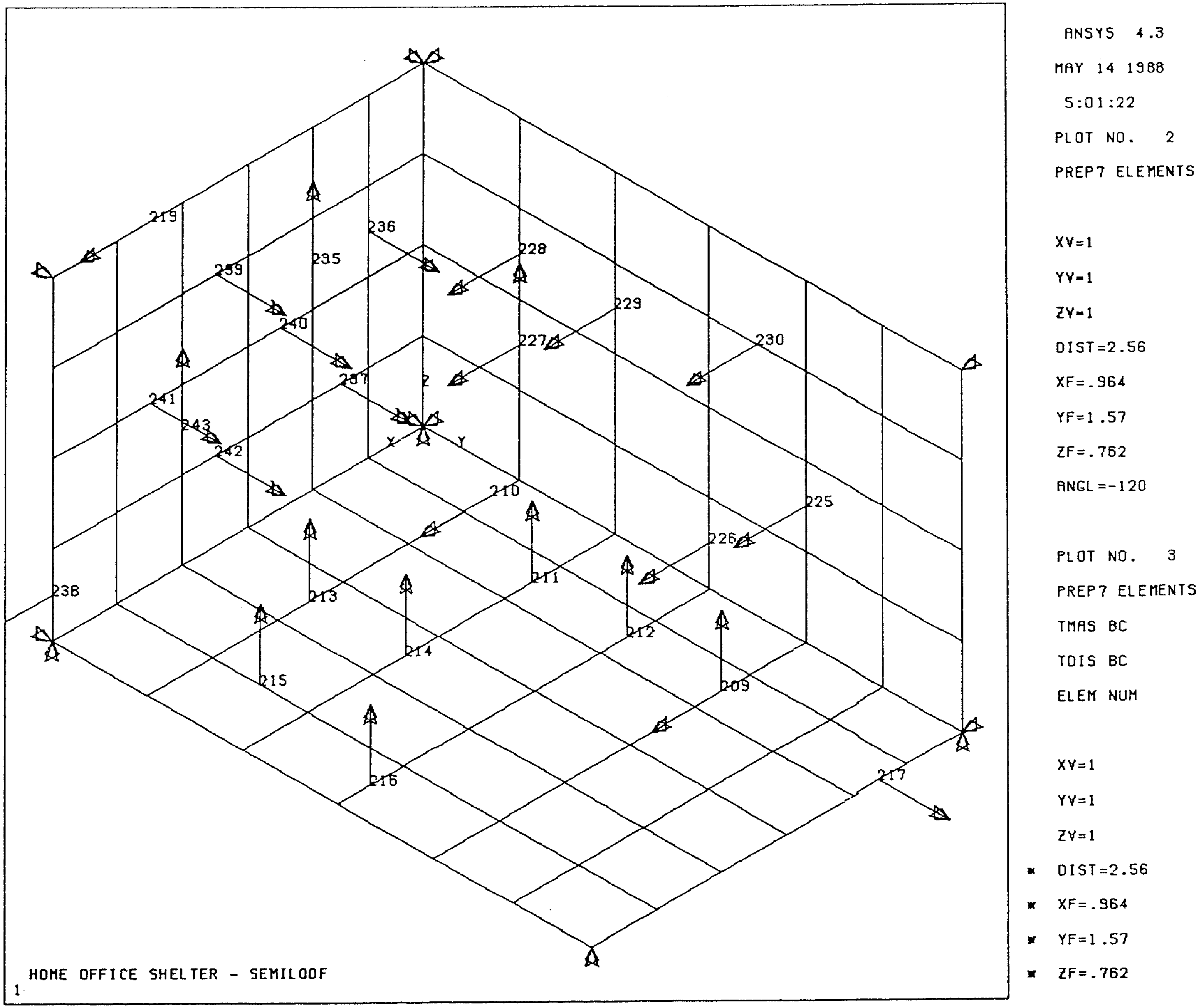


Figure 2.9 Dynamic degrees of freedom.

CHAPTER 3

DYNAMIC ANALYSIS OF A CONCRETE NUCLEAR SHELTER USING SEMILOOF SHELL ELEMENTS IN ANSYS

3.1 Introduction

A breakdown of the energy distribution from an above ground nuclear detonatiion is shown below:

45% Blast and shock waves.

35% Light and heat.

5% Initial nuclear radiation.

5% Residual radiation and fallout.

A structural analysis is normally concerned only with the energy released in the form of blast and shock.

After detonation of the device, a circular pressure wave is propogated outwards from the centre of blast, as shown in Fig.3.1. The air behind the pressure front is compressed and an 'overpressure' is applied to any structure it encapsulates. Because the pressure front is moving, an additional dynamic (wind) loading is also applied to structures in its path. The semi-buried nuclear shelter considered in this thesis will only be subject to the overpressure effects by virtue of the fact that it is underground and out of the path of an advancing pressure front.

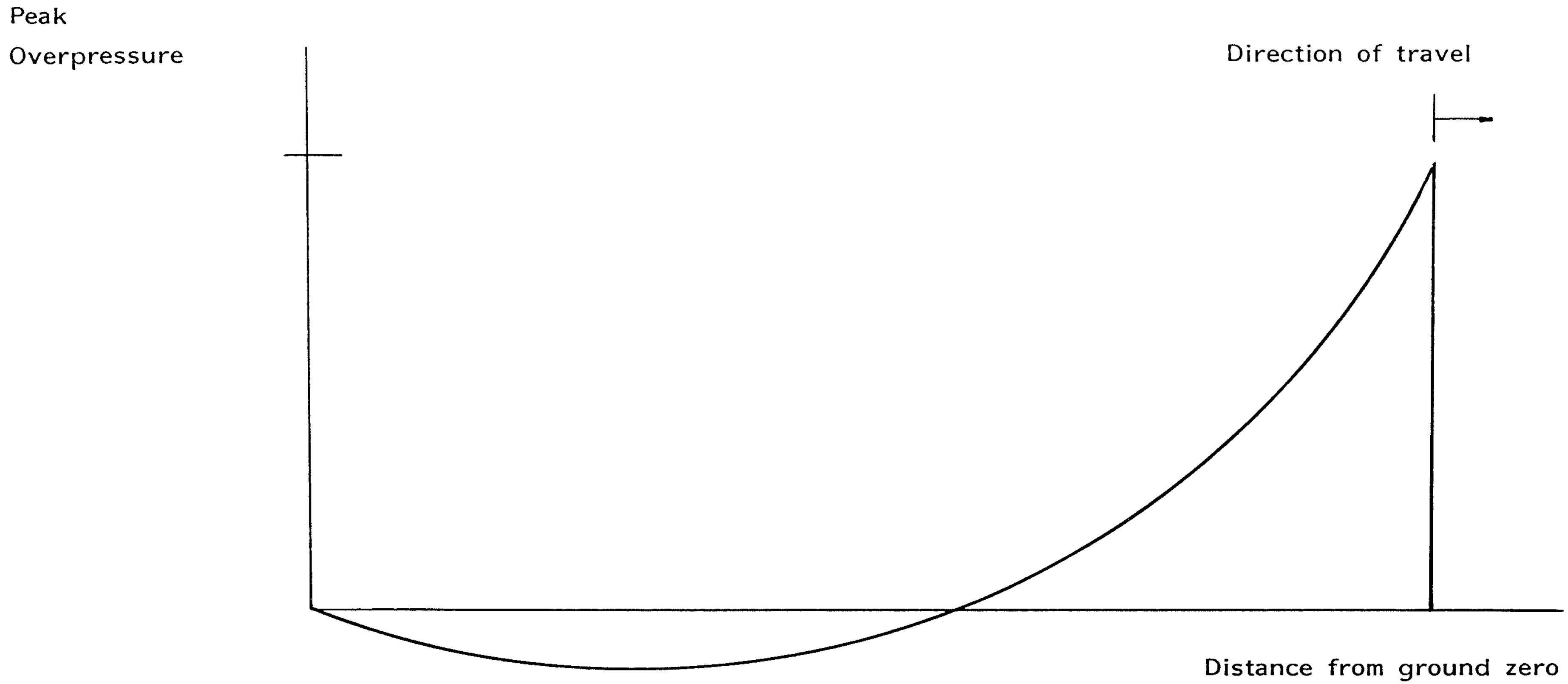


Figure 3.1 Variation of overpressure with time at a fixed location.

3.2 Shelter analysis methodology

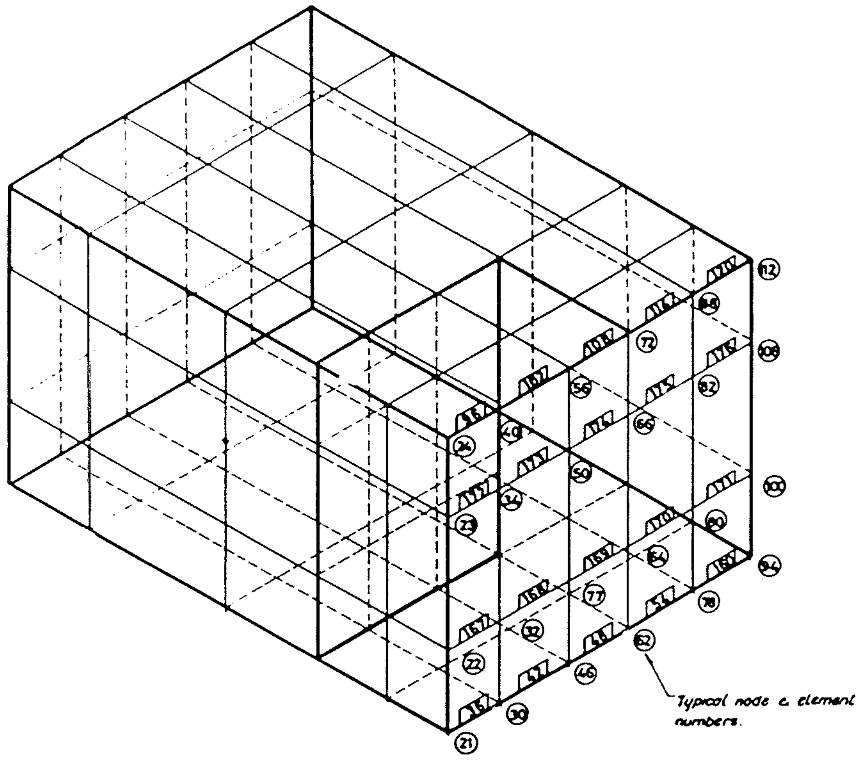
The Home Office guide suggests a design based on results of various static analyses carried out on a six person domestic nuclear shelter. The finite element method was used for all the analyses. An elementary beam model with an elastic foundation was used for the static analysis, and a reduced lumped mass model with foundation damping for the dynamic analysis. The finite element meshes used by the Home Office working party are shown in Fig.3.2.

In order to verify the results of the analyses published in the Home Office guide, the analyses will be repeated in this thesis and, where possible, a direct comparison of results will be made. In order to acheive a greater accuracy, a more refined mesh of continuem SemiLoof shell elements will be used. Both static and transient analyses will be carried out to ensure that the design criteria are met under time dependant conditions, though in the latter case no direct comparison of results can be made as those of the working party are not available.

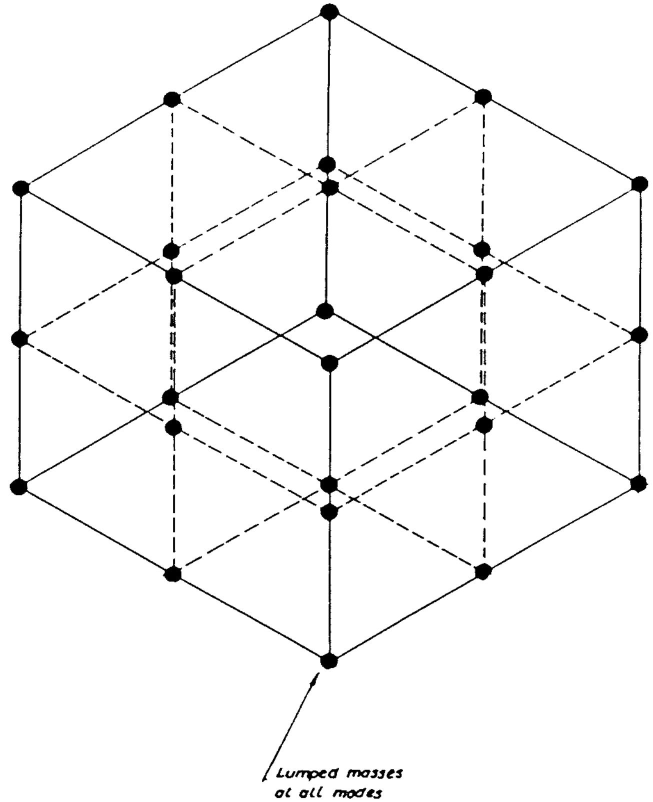
The design loadings for the analyses are summarised in Fig.3.2.

A peak overpressure of 1.0 atmosphere (103.350 kN/m^2) was applied to the roof and within the stairwell, and an overpressure of 0.5 atmospheres (51.675 kN/m^2) was applied to the outside of all walls. This design load describes that expected to be seen by a nuclear shelter, semi-buried in dry ground, subject to a 10Mt blast load at a distance of 7km (11).

It is assumed that the load is applied 'instantaneously' and will decay with time according to the equation (12):



STATIC MODEL



SIMPLIFIED DYNAMIC MODEL

NOTATION:

(21) Indicates node numbers.

Indicates element number.

APPLIED LOADS

1 Self wt of structure.

2 Paving surcharge.

3 Active soil pressure.

4 Overpressure of 1 atmosphere (1 bar or 15 p.s.i.) on roof and within stairwell.

5 Overpressure of ½ atmosphere (0.5 bar or 7.5 p.s.i.) on outside of all walls.

APPLIED LOADS

1. Overpressure of 1 atmosphere (15psi) on roof.

2. Overpressure of ½ atmosphere (7.5 psi) on outside of all walls.

3. Applied pressure curves.

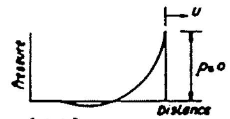


Fig.1

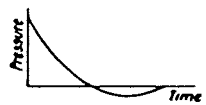


Fig. 2

Figure 3.2 Home Office beam models. (11)

AMENDMENTS DATE

CONTRACTOR MUST VERIFY ALL DIMENSIONS ON SITE BEFORE COMMENCING ANY WORK OR MAKING ANY SHOP DRAWINGS

CIVIL AND STRUCTURAL ENGINEERS SECTION

CHIEF ARCHITECT & DIRECTOR OF WORKS HOME OFFICE LONDON

DESIGNED	H S
DRAWN	. B
CHECKED	

STRUCTURAL REFERENCE NUMBER

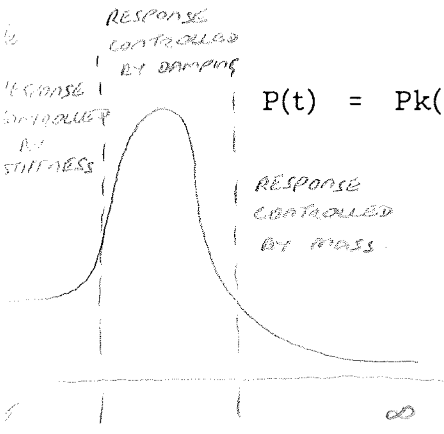
SCALES

SHELTER STUDIES

6 PERSON FAMILY SHELTER STATIC & DYNAMIC COMPUTER MODELS

FILE NO | SPEC NO | DRAWING NUMBER

DATE



$$P(t) = Pk(1 - t/T)\, e^{-(t/T)} \qquad (3.1)$$

where $P(t)$ is the overpressure at time t

Pk is the peak overpressure

t is the current time

T is the total time of the load

The term 'instantaneously', in the present context, means of a duration, relative to the response frequency of the structure, which causes the response to be controlled by the structure mass. (See fig. 3.2a) This concept determines whether or not a dynamic analysis is appropriate and is discussed further in Section 4.3.

3.2.1 Static analysis

The first analysis in this thesis conducted on the Home Office shelter was a static analysis which served to check that the distribution and magnitude of the results obtained were similar to those given in (11). The static analysis also served to check the mathematical stability of the model. i.e. to check for ill-conditioning or negative diagonal terms which may have occured in the stiffness matrix. The static pressure loadings, shown in Figs. 3.3 to 3.4 were (11):

Static pressure on roof and internal stairwell of 103.350 kN/m².

Static pressure on vertical walls of 51.675 kN/m².

3.2.2 Transient analysis

Before the analysis could be conducted in the time domain, information was first obtained about the dynamic characteristics of the model in order to make decisions about the duration of the 'instantaneous' load, the reduced duration of the applied load, the size of the integration time step and parameters for the

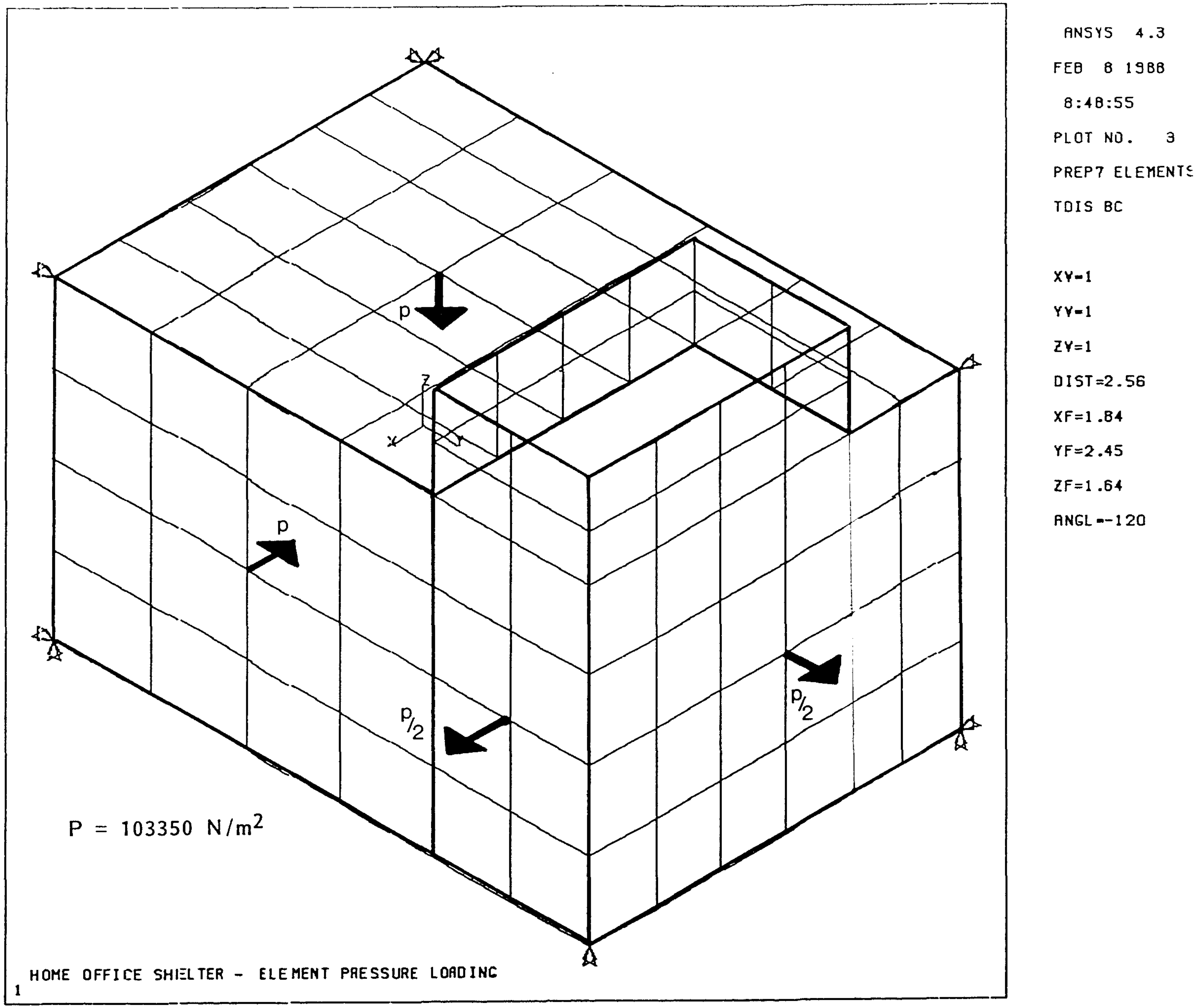


Figure 3.3 Overpressure loading on nuclear shelter.

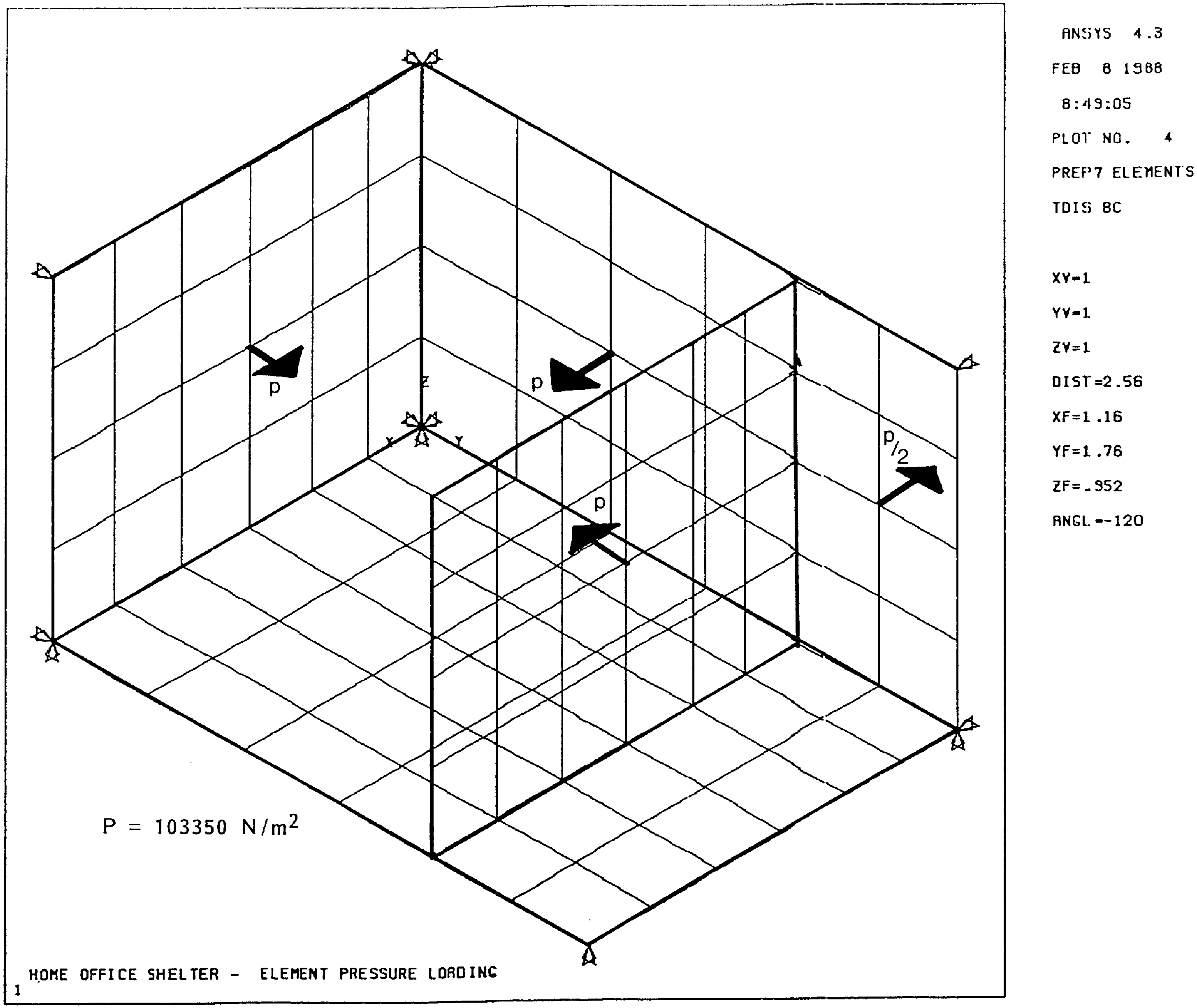


Figure 3.4 Overpressure loading on nuclear shelter.

calculation of structural damping.

All this information was gleaned from the modal analysis which calculated the natural frequencies and mode shapes of the structure. From the results of the modal analysis, sufficient information was generated to enable the calculation of data required for the transient analysis. Hence the data preperation for the transient is detailed in Chapter 4 after the presentation of the modal analysis results.

3.2.3 Damping characteristics

Damping is one of the more difficult features of a transient analysis to evaluate and it manifests itself in the reduction of response amplitudes and the changing and phasing of response frequencies.

In this thesis, damping has been defined in the form of Rayleigh constants which simulate strain energy damping, or 'dry' damping, which is material dependant (1). The damping constants act as factors to the mass and stiffness matrices:

$$[C] = a\,[M] + b\,[K] \tag{3.2}$$

where [C] is the structure damping matrix

[M] is the structure mass matrix

[K] is the structure stiffness matrix

The two Rayleigh constants representing mass and stiffness damping are functions of a constant damping ratio; the ratio of critical to actual damping. The relationship between damping constants is:

$$\zeta = a/2\omega + b\omega/2 \qquad (3.3)$$

where ζ is the constant damping ratio

a is the frequency dependant mass damping factor

b is the frequency dependant stiffness damping factor

ω is the response frequency of interest (Rads/sec)

In order to solve for a and b, two response frequencies from the modal analysis were selected, (the highest and lowest of interest, determined from the modal analysis results), which enabled two simultaneous equations to be defined from eqn. 3.2,which were then solved for a and b.

Hence $a = 4*\pi*\zeta(1/f1 - 1/f2) / (1/f1^2 - 1/f2^2)$ (3.4)

and $b = \zeta(f2 - f1) / \pi(f2^2 - f1^2)$ (3.5)

where f1 = lowest response frequency of interest (Hz)

f2 = highest response frequency of interest (Hz)

The damping ratio is assumed to be constant over a response frequency range whilst a and b vary with frequency. Fig.3.5.

The constant damping ratio 'ζ' was assumed to be 0.05 for concrete.

No foundation damping has been included as this will allow the structure to vibrate more freely and thus induce greater stresses.

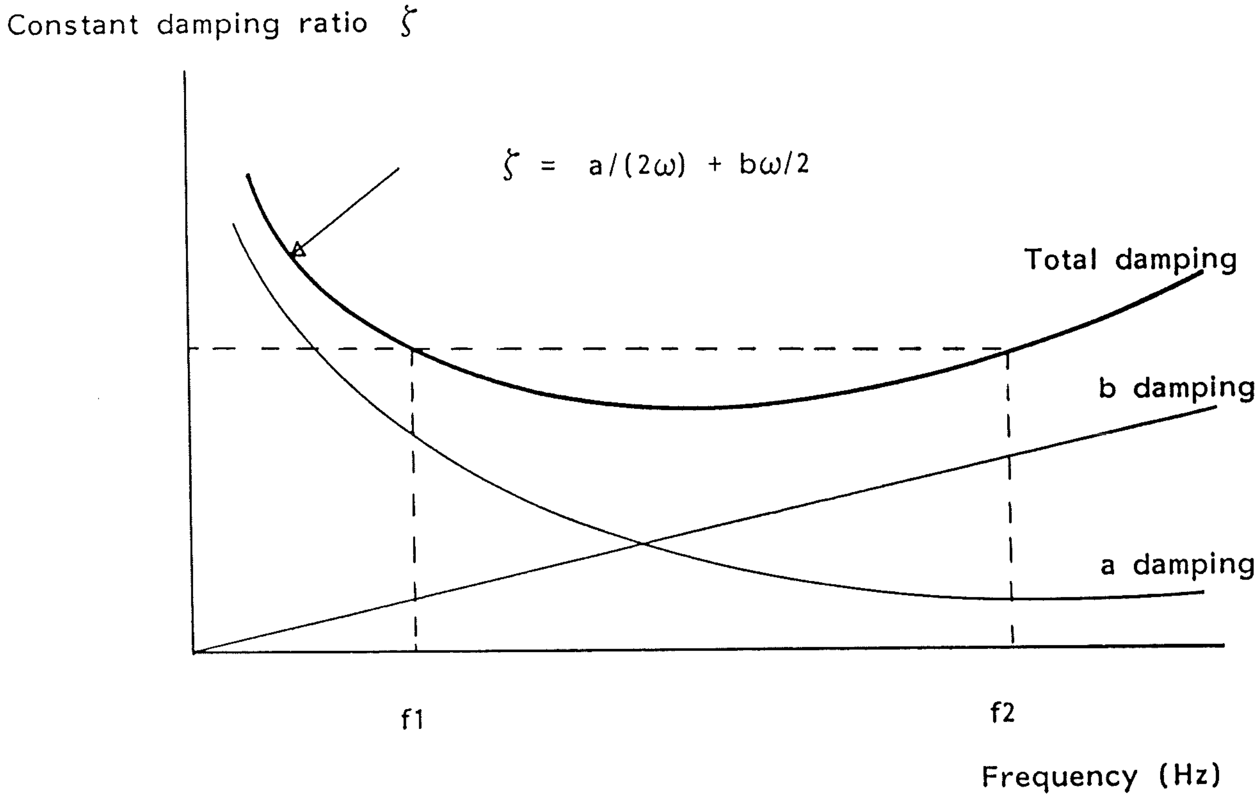


Figure 3.5 Relationship between damping constants

CHAPTER 4

RESULTS EVALUATION OF THE SEMILOOF SHELTER ANALYSIS

4.1 Introduction

In this thesis, three analyses have been conducted on the nuclear shelter:

a. Static analysis

b. Modal analysis

c. Transient analysis

The results of each of the three analyses will be considered in turn and information needed to calculate factors appropriate for further analyses are discussed.

4.2 Static analysis

The static analysis served to check that the results obtained from the model used in this thesis correlated with the results published in the Home Office guide (11).

Figs. 4.4 to 4.12 show the displacement and bending moment distributions for the static load in the SemiLoof elements. The static analysis of the shelter was repeated with ANSYS Stif93 eight noded thin shell elements for the sake of verification of the SemiLoof results. The results of the Stif93 analysis are presented in Figs. 4.14 to 4.22 and show good agreement with those calculated by SemiLoof.

It is important to note that the averaging of moments which is required to produce contouring can give misleading values, especially at boundaries between elements without common orientation. For this reason all magnitudes quoted are unaveraged centroidal values. The element axes are shown in Figs.4.1 to 4.3.

Even with averaged values it can be seen that there is close correlation between the moments depicted in Figs.4.7 to 4.12, and those calculated by the Home Office working party, Fig.4.13.

A maximum moment of 33 kNm/m has been calculated in element 197 in the blast wall which agrees with that calculated in a similar position in the Home Office model. The bending moments from the Home Office analyses tend to be higher towards the edges of the walls. This may occur for the following two reasons:

a. The mesh used in the Home Office guide is a simple grillage, so the magnitudes of the bending moments at these points may tend to be generous due to the greater flexibility of beams elements over continuem elements.

b. The moments illustrated in Figs.4.7 to 4.12 are centroidal moments, therefore the moment peaks at the wall edges have not been resolved and will need to be calculated by submodelling (1).

The process of submodelling aims to recover the bending moment peaks by refining the finite element mesh in the appropriate areas. Sections of the coarse model were re-modelled with a finer mesh, interpolating the displacements and rotations from the coarse model and using them as initial displacements and rotations for the fine model. The submodelling capability is not available for use with the user element so interpolation had to be done using the Stif93 model. The submodelling plots are Figs.4.23 to 4.30. The elements involved in the fine mesh were uniformly orientated to yield more accurate moment averaging. These plots show that the peak moments in the sections chosen are 40kNm/m at the top of the blast wall and 29kNm/m at the edge of the roof, which as expected, are higher than those of the coarse model.

By comparing the peak moment values at the element edges from the submodel with those of the coarse model, it can be seen that there is an increase in moment magnitude by a factor of 2.1. This fact will be borne in mind when evaluating the results of the transient analysis.

From Chapter 2, the ultimate moments of resistance for moderate damage are 90 kNm/m and 51 kNm/m in the blast wall and all other shelter panels respectively. These are much higher than the calculated moments of 40 kNm/m and 29kNm/m respectively and hence it has been shown that for the static loading case only, the design meets the design criteria. The transient responses will now be calculated.

4.3 Modal analysis

The purpose of the modal analysis was to determine the dynamic characteristics of the model and obtain further information useful to the transient analysis.

The results of the modal analysis are in terms of natural frequency and mode shape. The modes are calculated in order to determine parameters for the calculation of integration time step and damping values for the subsequent transient analysis.

Mode shape plots are presented in Figs.4.31 to 4.35 and represent the natural frequency responses in an undamped free vibration environment of the first ten natural frequencies of the model.

Table 4.1 shows the natural frequencies and the effective mass participating in those modes. By summing the effective masses, it can be seen that in all three orthogonal directions, over 80% of the structural mass participates in the first five modes. It is expected that the structure will respond to the impulsive blast load mainly in its three fundemental frequencies, 68 Hz in the X-direction, 78 Hz in the Y-direction and 63 Hz in the Z-direction, together with less important contributions from the other modes. After the fifth mode, the mass participations are at such a level that any response of the structure in these modes will be insignificant, and therefore, no special effort will be made to resolve them. The accuracy of resolution of the response modes in the transient analysis is determined by the integration time step. For verification purposes, the modal analysis was repeated with Stif93 elements, whose modes show good agreement with SemiLoof (Figs.4.36 to 4.40).

4.4 Transient analysis

The transient analysis was the major analysis conducted on the shelter in order to measure its response to the blast load.

The structure mass was defined in the same way as for the modal analysis, by lumping the mass at nodal degrees of freedom which characterize the low frequency modes.

4.4.1 Integration time step size

The results of the modal analysis have shown that the fundemental natural frequencies of the shelter are:

mode 1 63 Hertz, period = 0.016 secs

mode 2 68 Hertz, period = 0.015 secs

mode 3 78 Hertz, period = 0.013 secs

mode 4 98 Hertz, period = 0.010 secs

mode 5 100 Hertz, period = 0.010 secs

The highest response frequency of interest is 100 Hz, therefore from (1):

Integration time step = period/20 = 0.01/20 = 0.0005 secs

It is assumed that the critical stage of the analysis is the time when the shock front impacts the ground above the shelter and thus the time domain included in the analysis will be limited to the range 0.0 to 0.1 seconds, by which time it is assumed that the structure will have suffered its highest loadings and deformations. This may be confirmed when evaluating the results of the transient analysis.

The total number of iterations required:

NITTER = total time/time step size

= .1/.0005

= 200 iterations.

Three continuous load steps will be used to apply the transient pressure loading:

Step 1, time = 0, initialising load step, No. of iterations = 1

Step 2 , time = 0 to 0.0005 secs, No. of iterations = 1

Step 3 , time = 0.0005 to 0.1 secs, No. of iterations = 199

4.4.2 Damping characteristics

Structural damping will be included in the form of Rayleigh constants which are dependant on the response frequency range of interest. The response frequency range of interest has been identified as 63 to 100 Hertz.

Therfore, using eqns.(3.4 and 3.5):

Damping constants for concrete a = 24.28 , b = 0.00010

for steel a = 14.57 , b = 0.00006

4.4.3 Time-history loading curve

The general shape of the curve, from peak overpressure back to atmospheric is defined by equation (3.1):

$$P(t) = P\ (1 - t/T)\ e^{-(t/T)}$$

Data for P and T in (3.1) is given in (11,12) and are relative to the bomb yield and the distance from ground zero.

An instantaneous pressure loading has been assumed which approximates to a rise time of 0.0005 secs. (1 iteration).

Hence, the loading curve for a 10 MT detonation at a distance of approximately 7 km from ground zero is given below, (11,12):

Peak overpressure	=	103350 N/m^2
Rise time	=	0.0005 secs
Total duration	=	5.0 secs

From (3.1), the pressure/time curve used in this analysis is shown in Fig.4.41.

4.4.4 Transient analysis results evaluation

The main analysis conducted on the nuclear shelter was the linear transient analysis. Figs.4.42 and 4.43 show the maximum displacement responses of the degrees of freedom shown in Figs.4.44 to 4.46, with time. It is evident from these responses that the peak bending moments will have occured in the first 0.04 seconds of excitation. The presence of the structural damping is responsible for the decay in amplitude with time.

Figs.4.42 and 4.43 also show that the shelter is responding in it's fundemental modes, identified in Section 4.3, which confirms the adequacy of the integration time step size for resolution of the modal responses.

It has been shown, (Sect.4.2) that the centroidal moments can increase by a factor of 2.1 towards the edges of the walls. It is assumed that this principle is also valid for the transient analysis, therfore all bending moments exceeding the ultimate moments / 2.1, were identified. These moments are listed in Table 4.2 and the elements involved shown in Fig.4.47. It is evident from Table 4.2 that the design criteria is likely to have been exceeded in elements 3 and 5, which are at the junction of the floor and the back wall. Elements 197 and 201 define part of the blast door frame, though it can be seen from Fig.4.48 that the moments decrease in magnitude towards the frame and hence it is appropriate to use the unfactored centroidal moments. The other elements in Table 4.2 are not at fixed panel edges and do not qualify for the increase factor. Hence their centroidal values can be used.
In summary, it appears that shelter meets the ultimate moment design criteria, except for elements 3 and 5 in the shelter floor as shown in Fig.4.47, which could be strenghtened by the addition of more reinforcing steel.

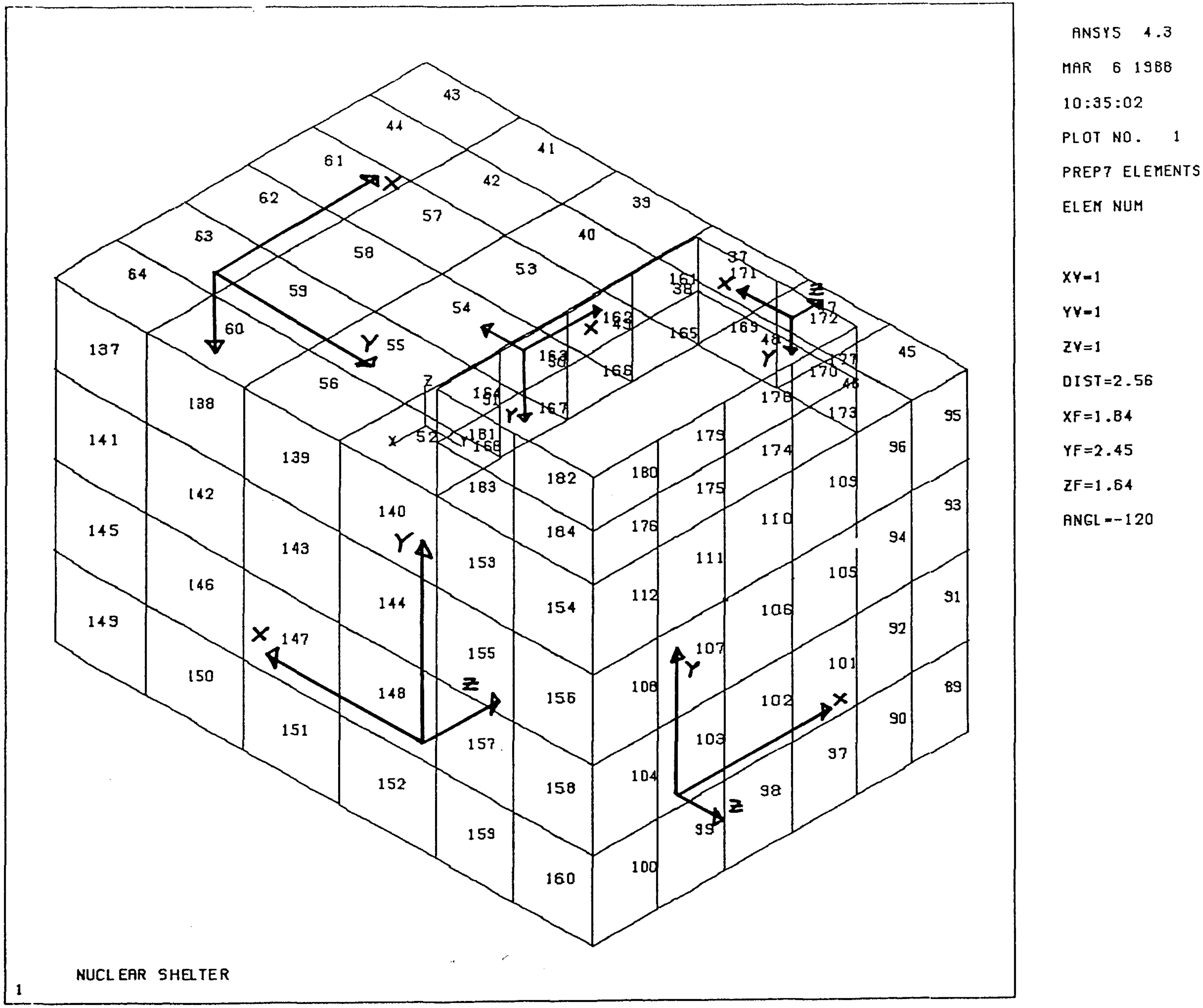


Figure 4.1 Element coordinate sytems axes.
Roof and front walls.

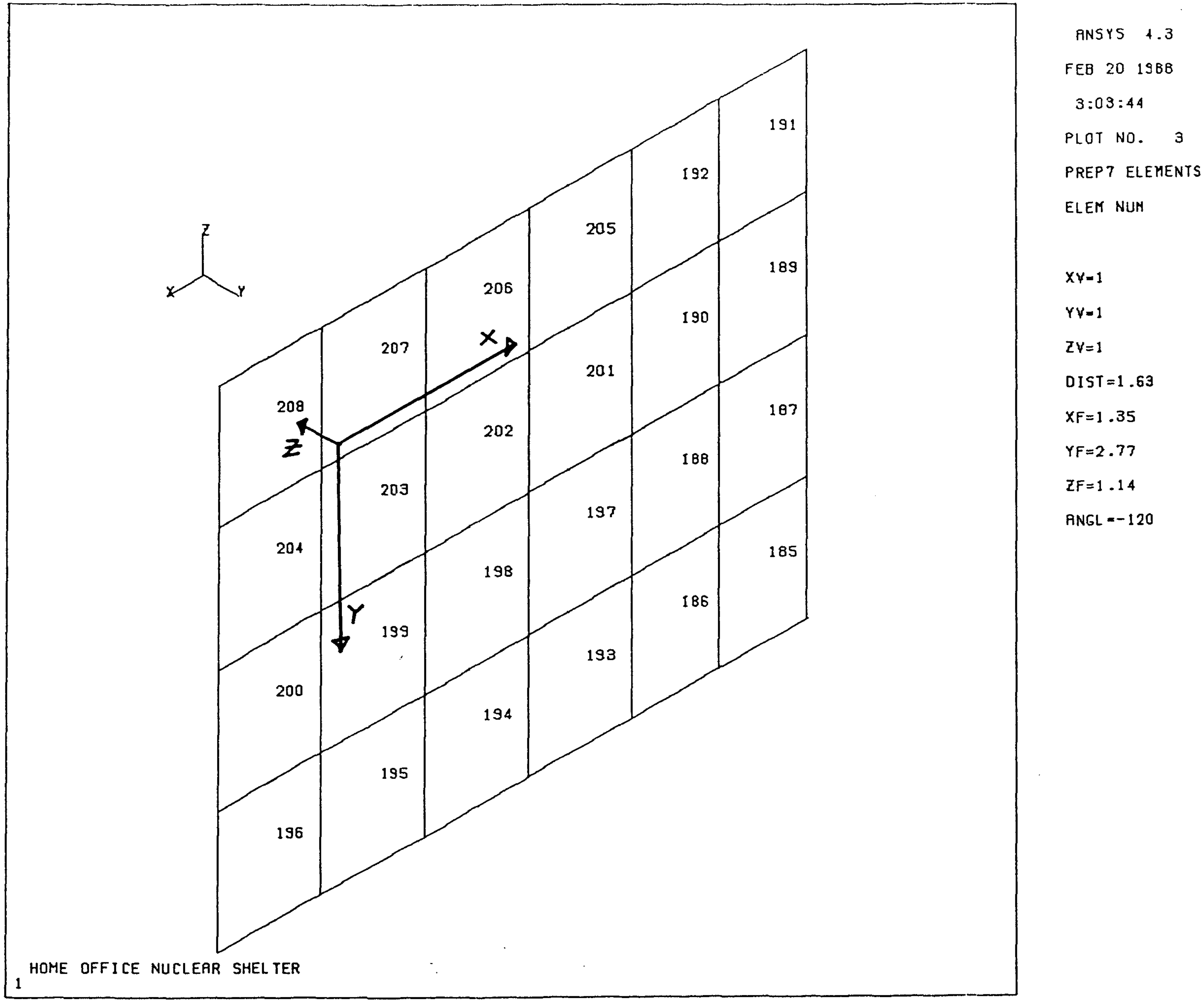


Figure 4.2 Element coordinate sytems axes.
Blast wall.

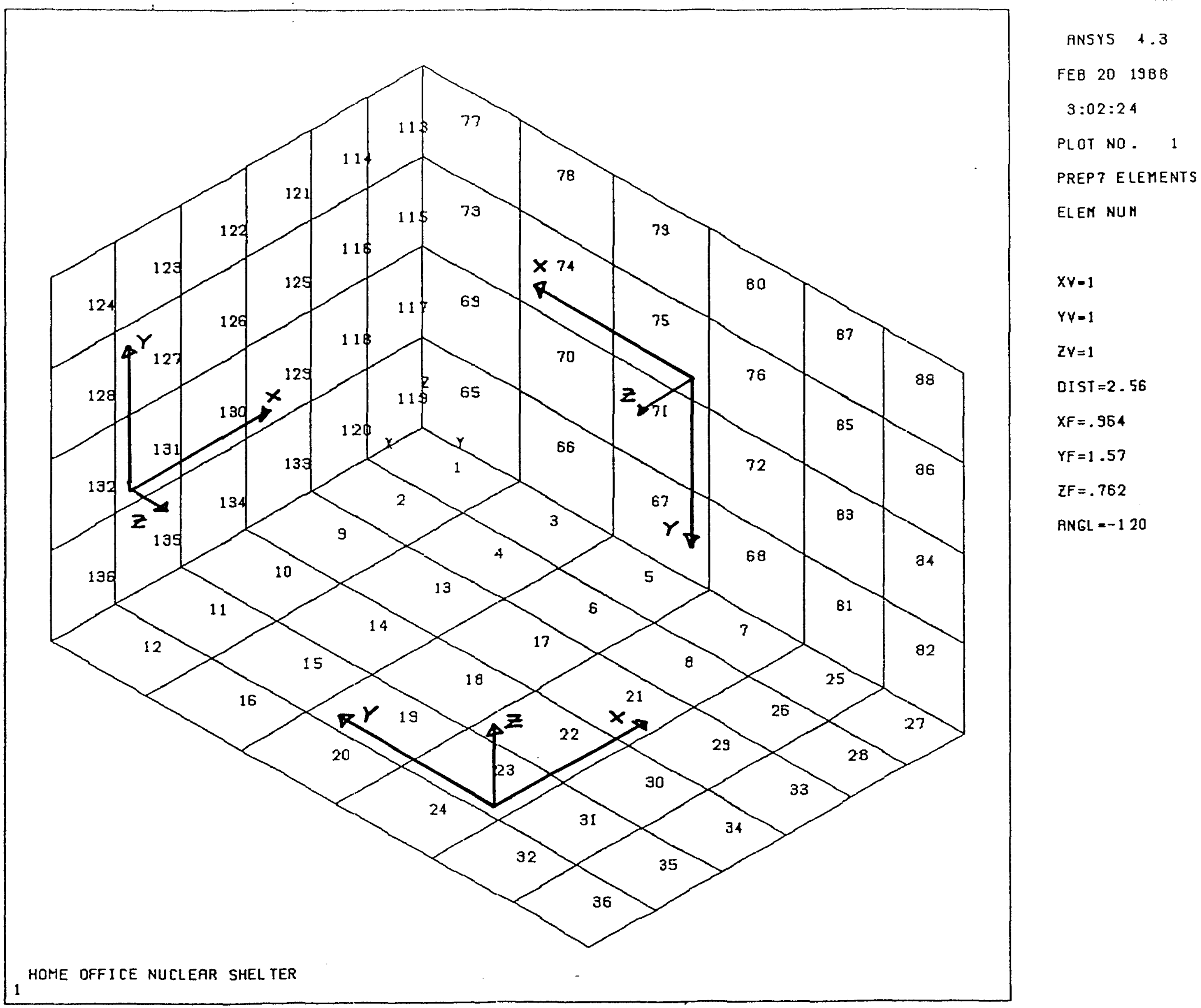


Figure 4.3 Element coordinate systems axes.
Floor and rear walls.

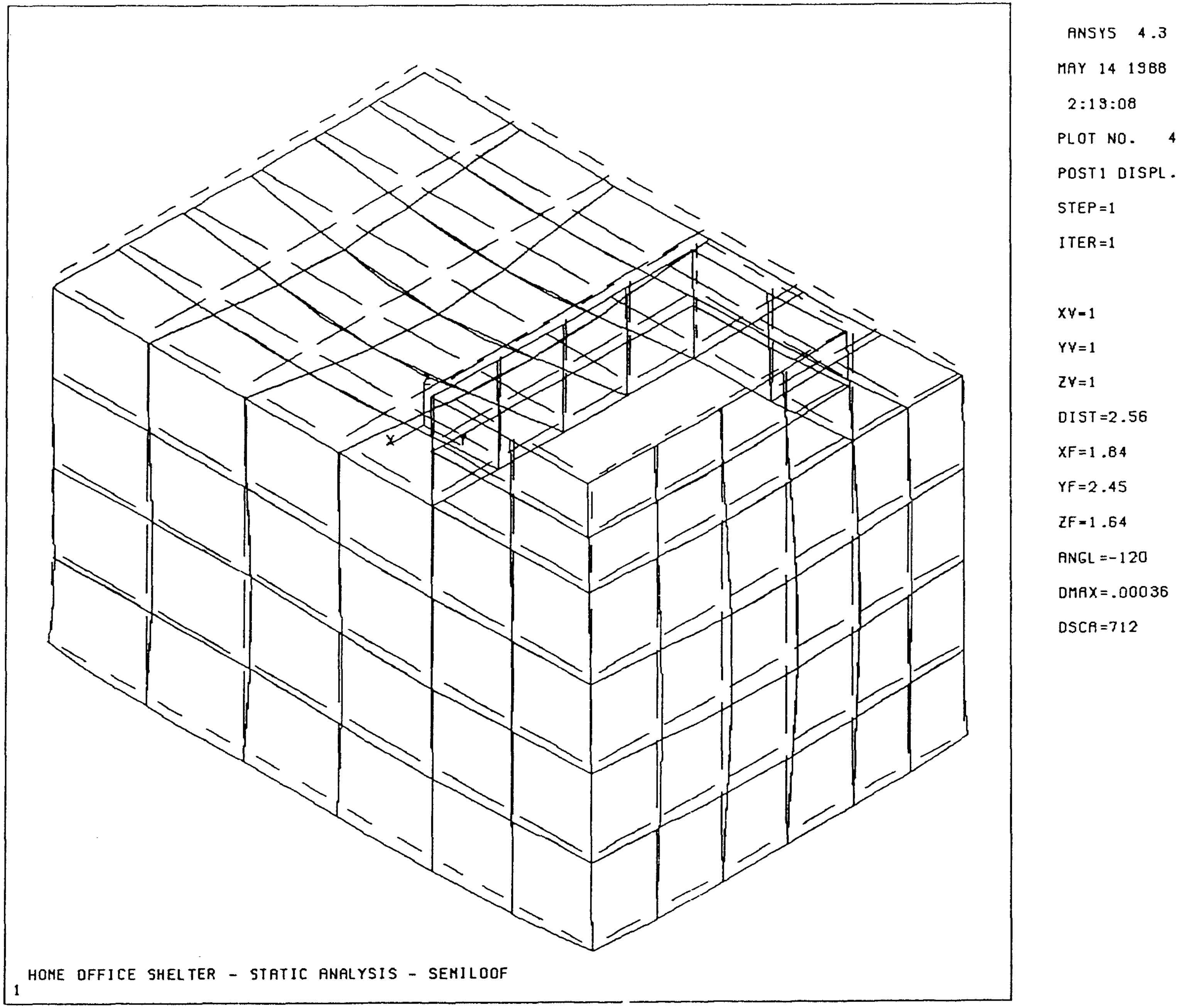


Figure 4.4 Static analysis displacement - SemiLoof Roof and front walls.

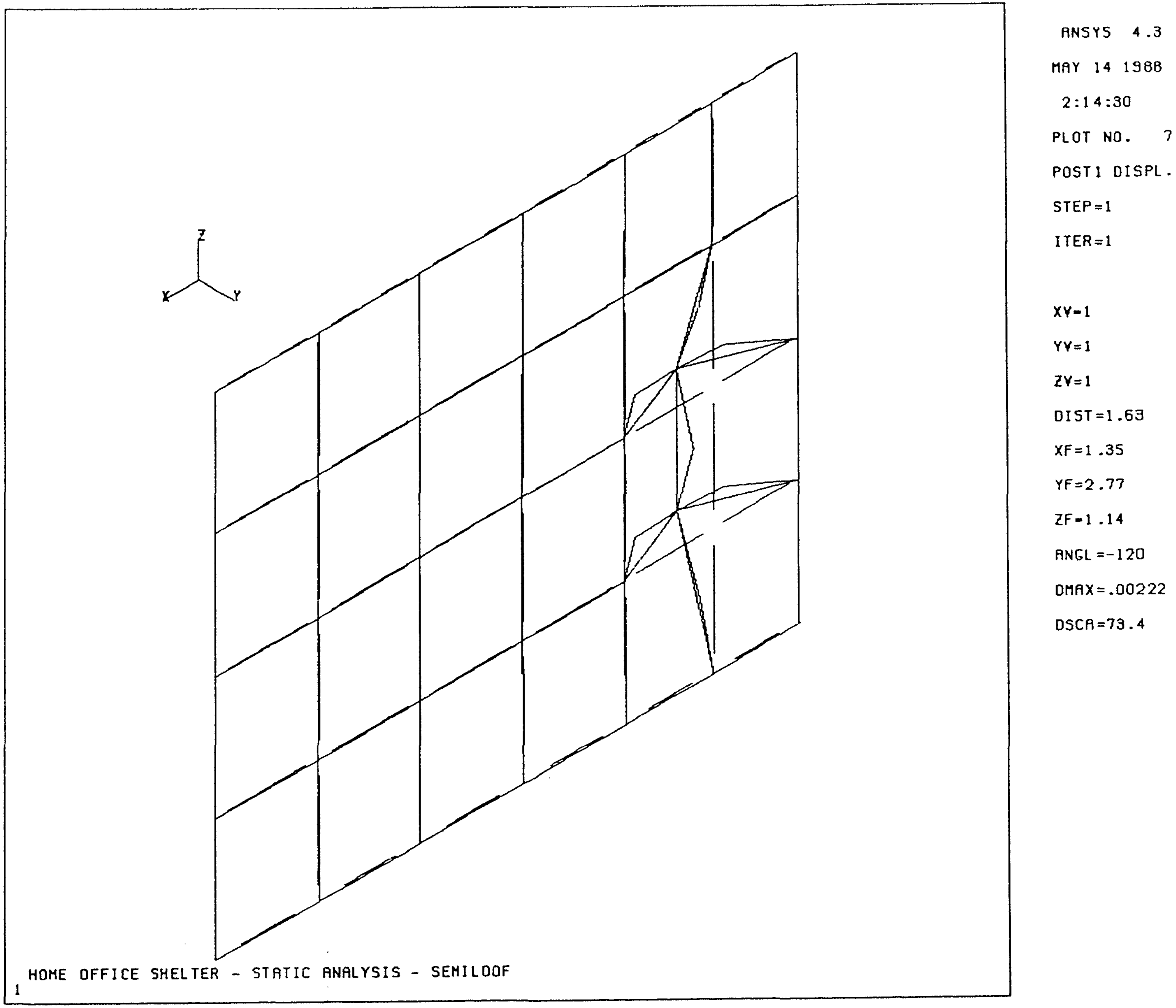


Figure 4.5 Static analysis displacement - SemiLoof.
Blast wall.

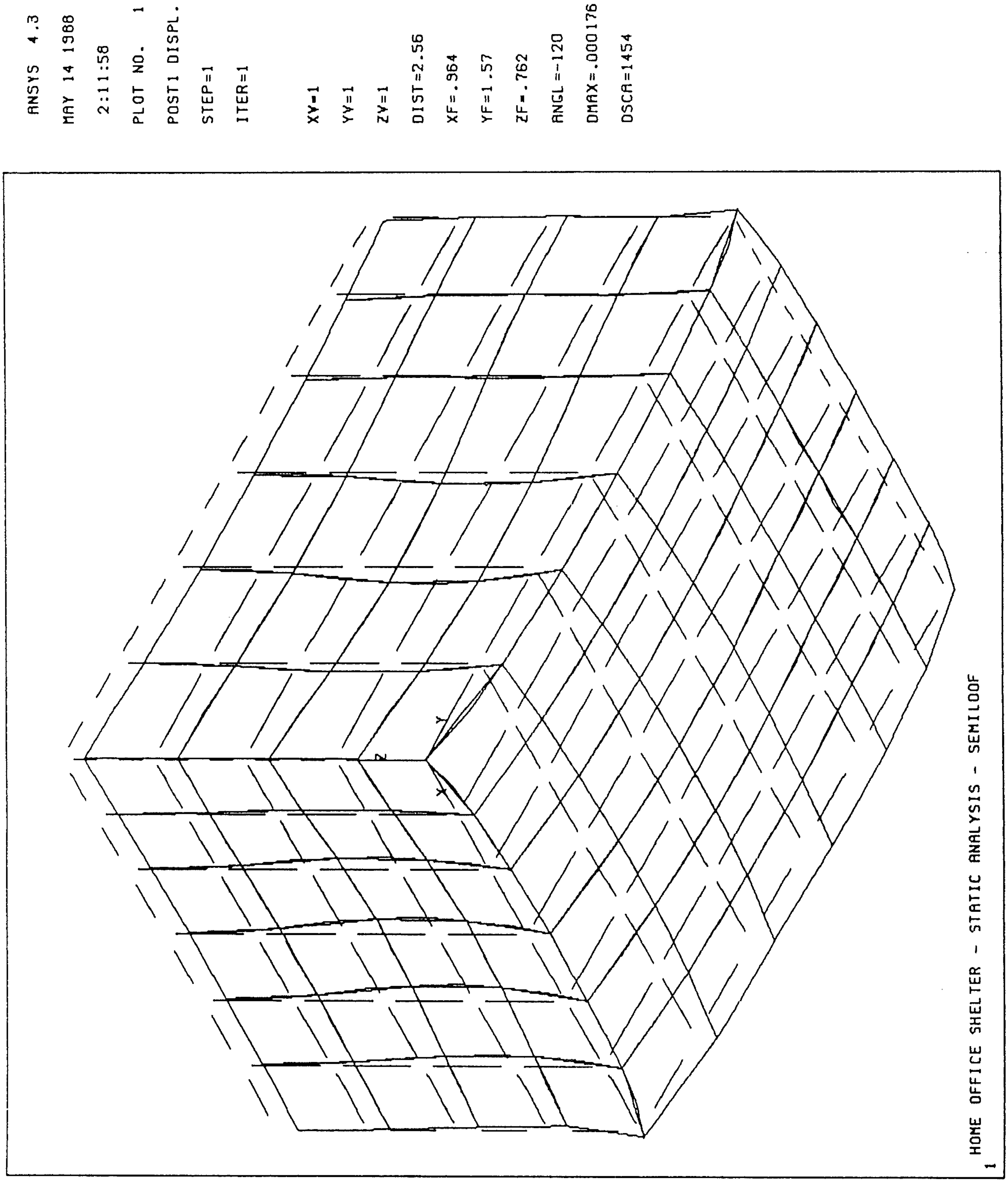


Figure 4.6 Static analysis displacement - SemiLoof.

Floor and rear walls.

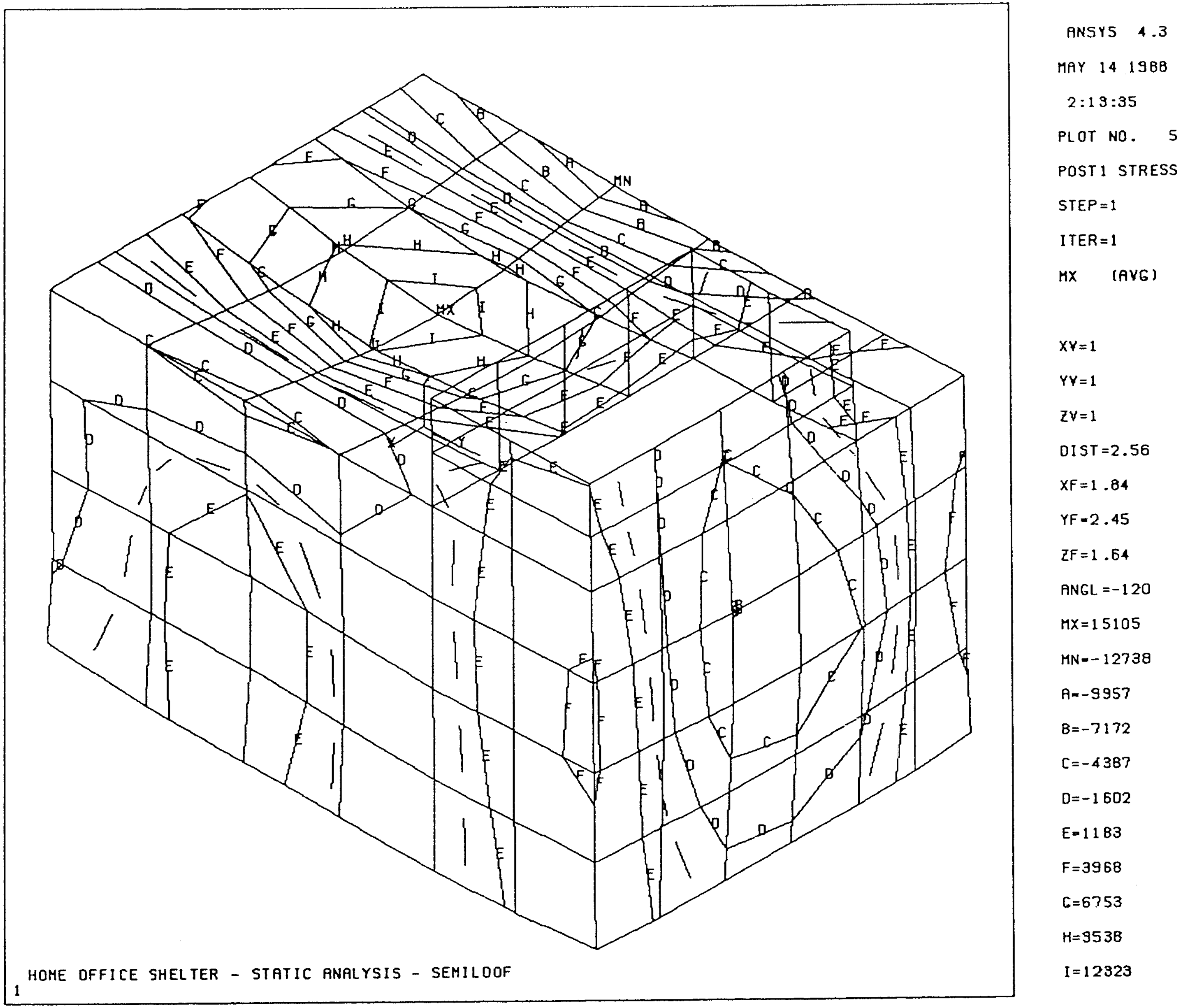


Figure 4.7 Static analysis bending moments - SemiLoof.
Roof and front walls. (MX)

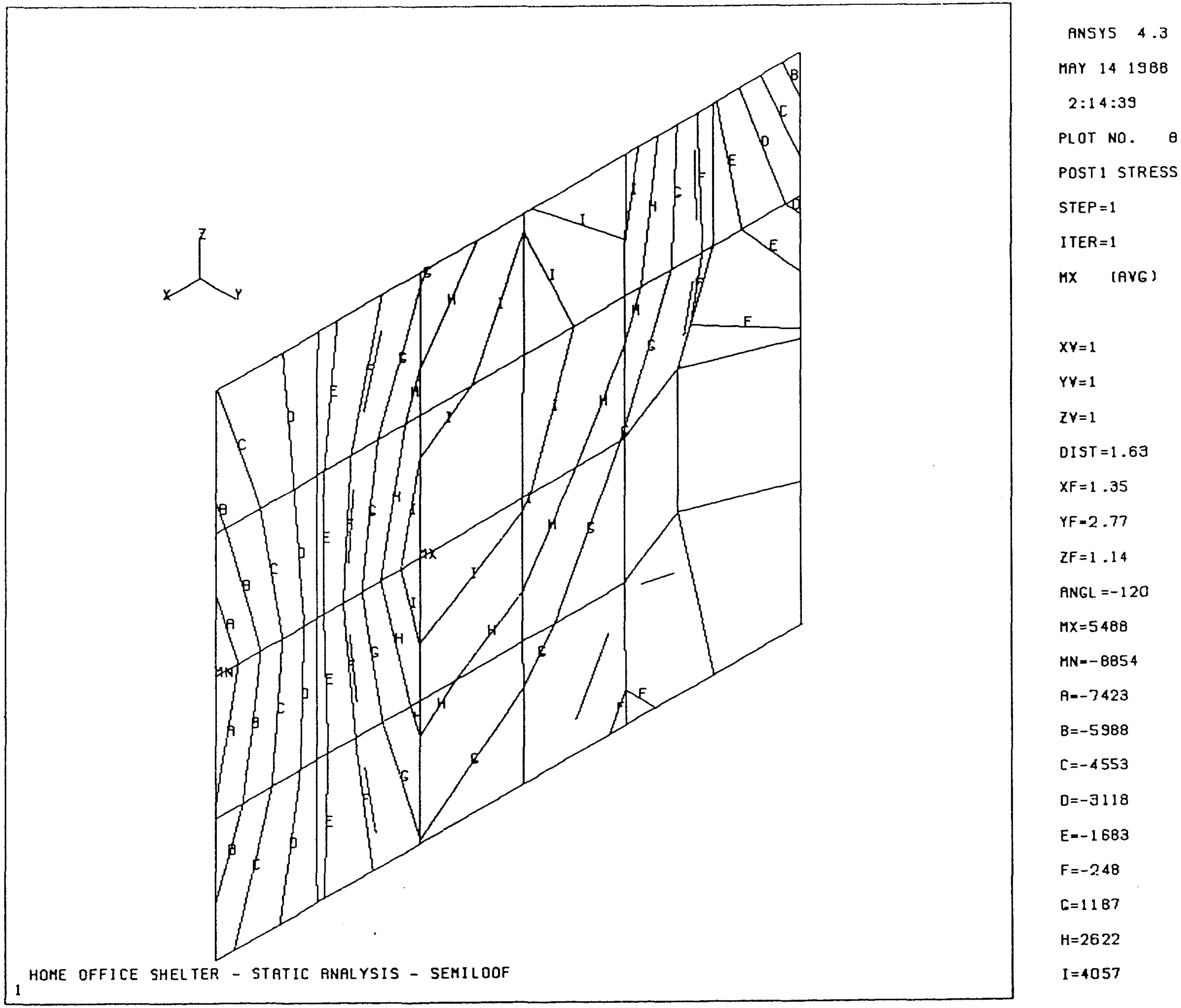


Figure 4.8 Static analysis bending moments - SemiLoof.
Blast wall. (MX)

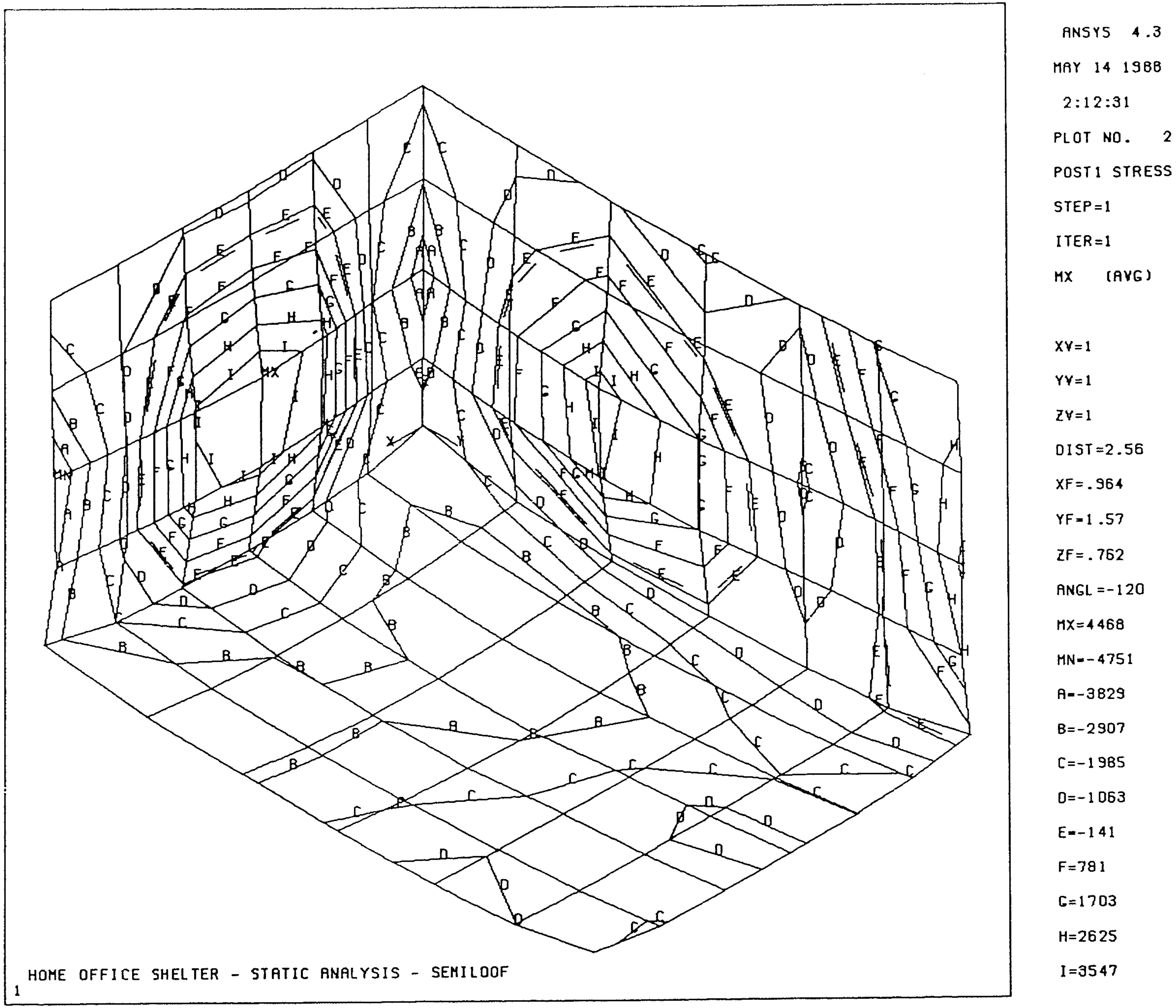


Figure 4.9 Static analysis bending moments - SemiLoof.
Floor and rear walls. (MX)

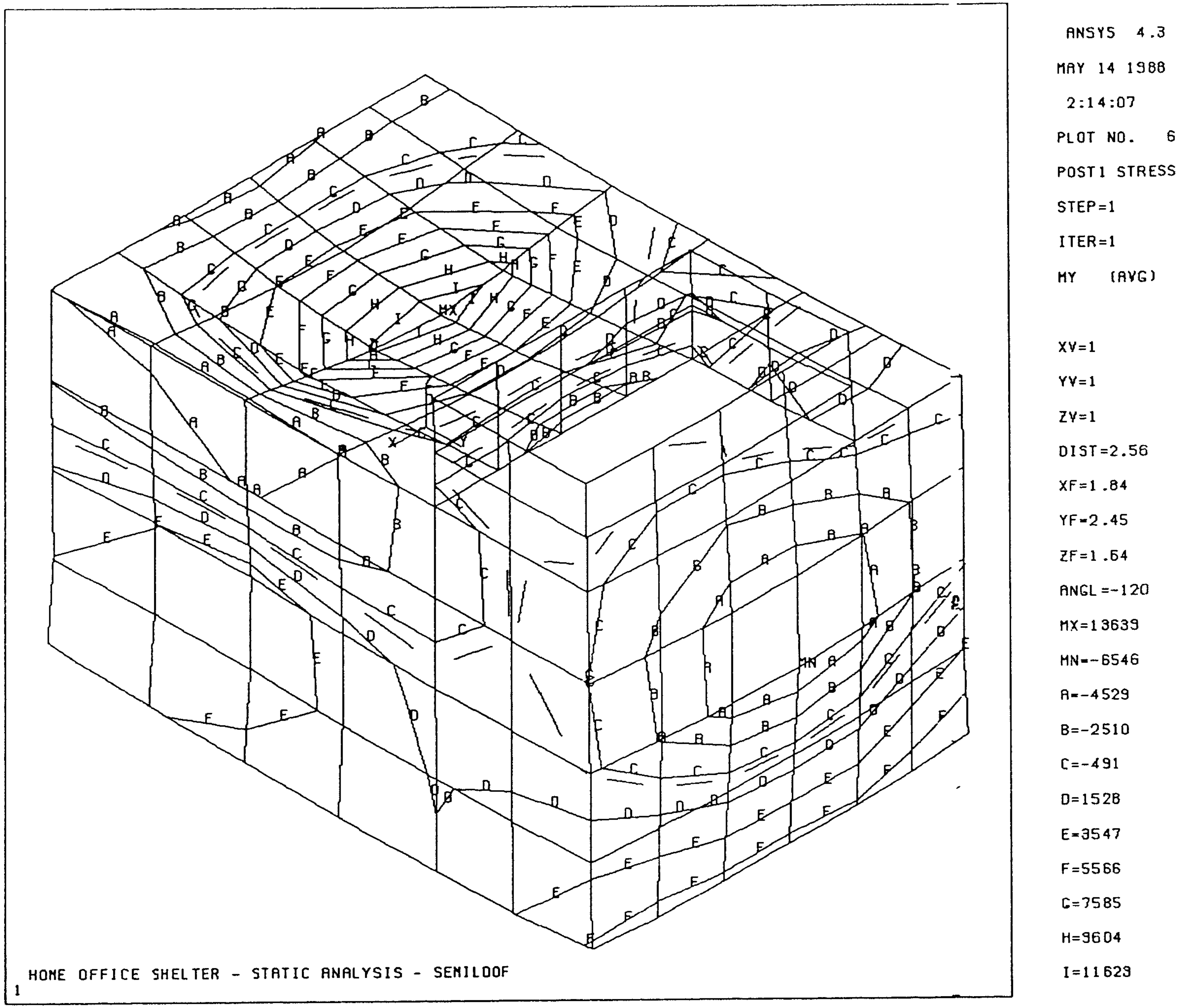


Figure 4.10 Static analysis bending moments - SemiLoof.
Roof and front walls. (MY)

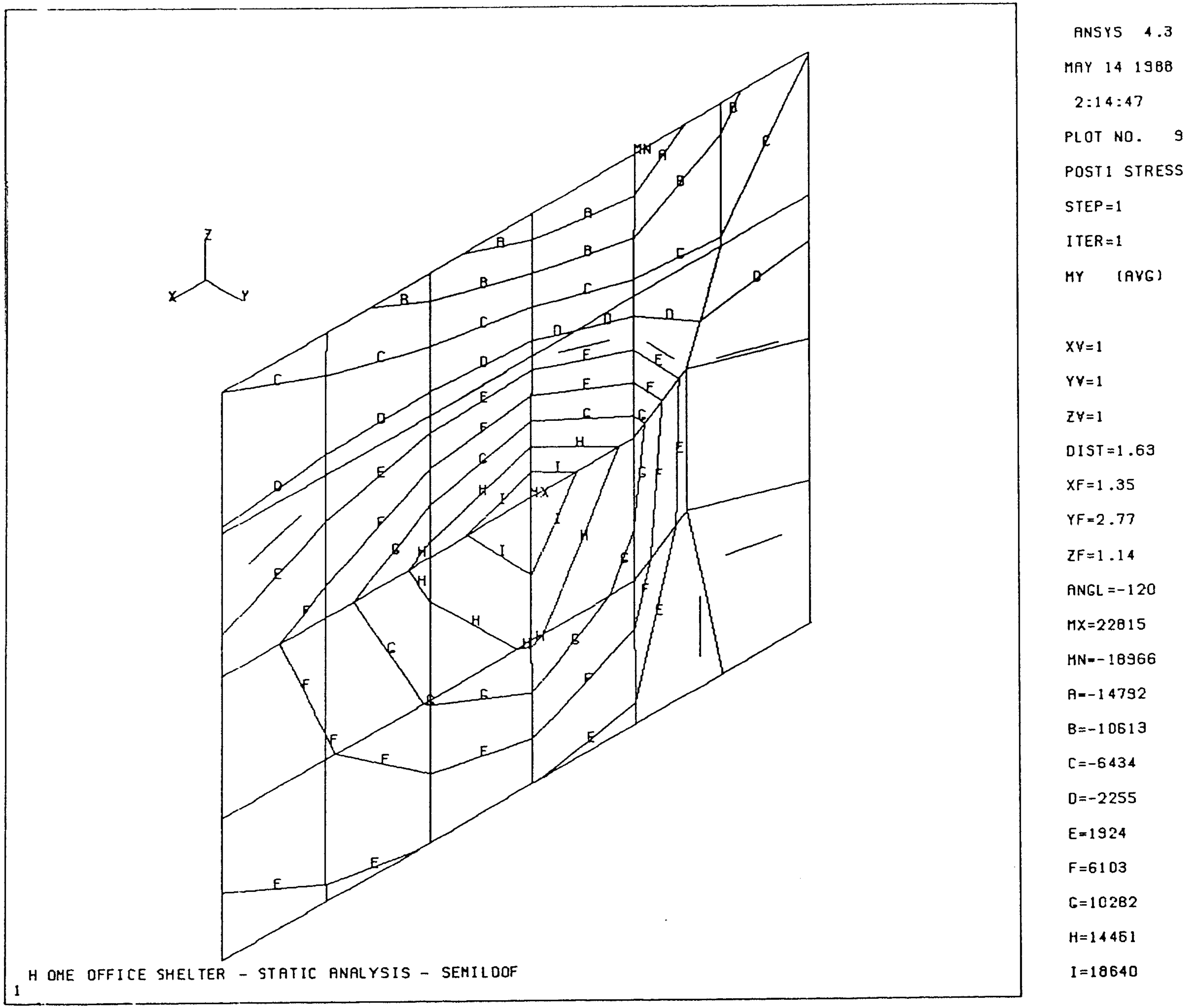


Figure 4.11 Static analysis bending moments - SemiLoof.
Blast wall (MY).

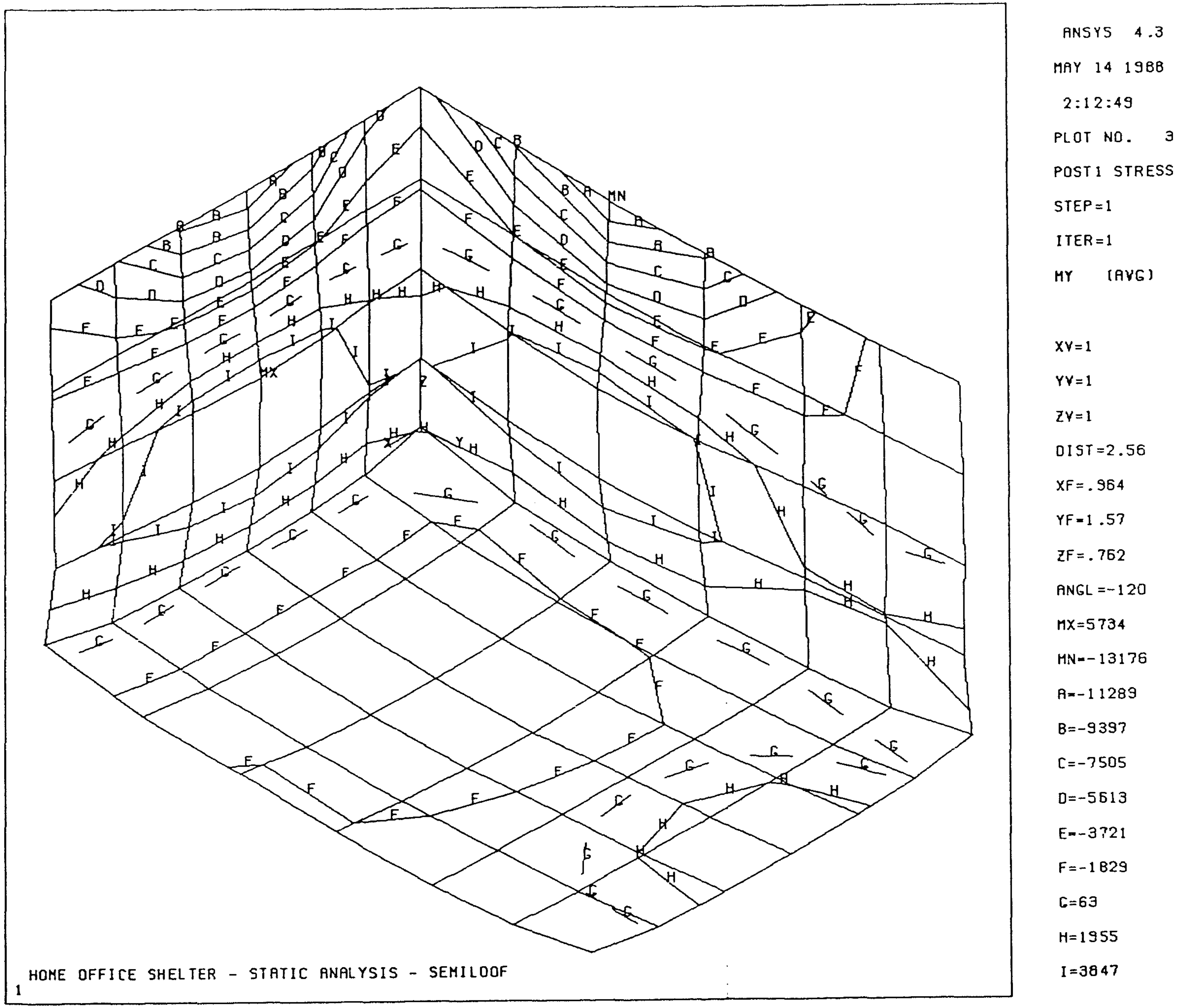


Figure 4.12 Static analysis bending moments - SemiLoof.
Floor and rear walls (MY).

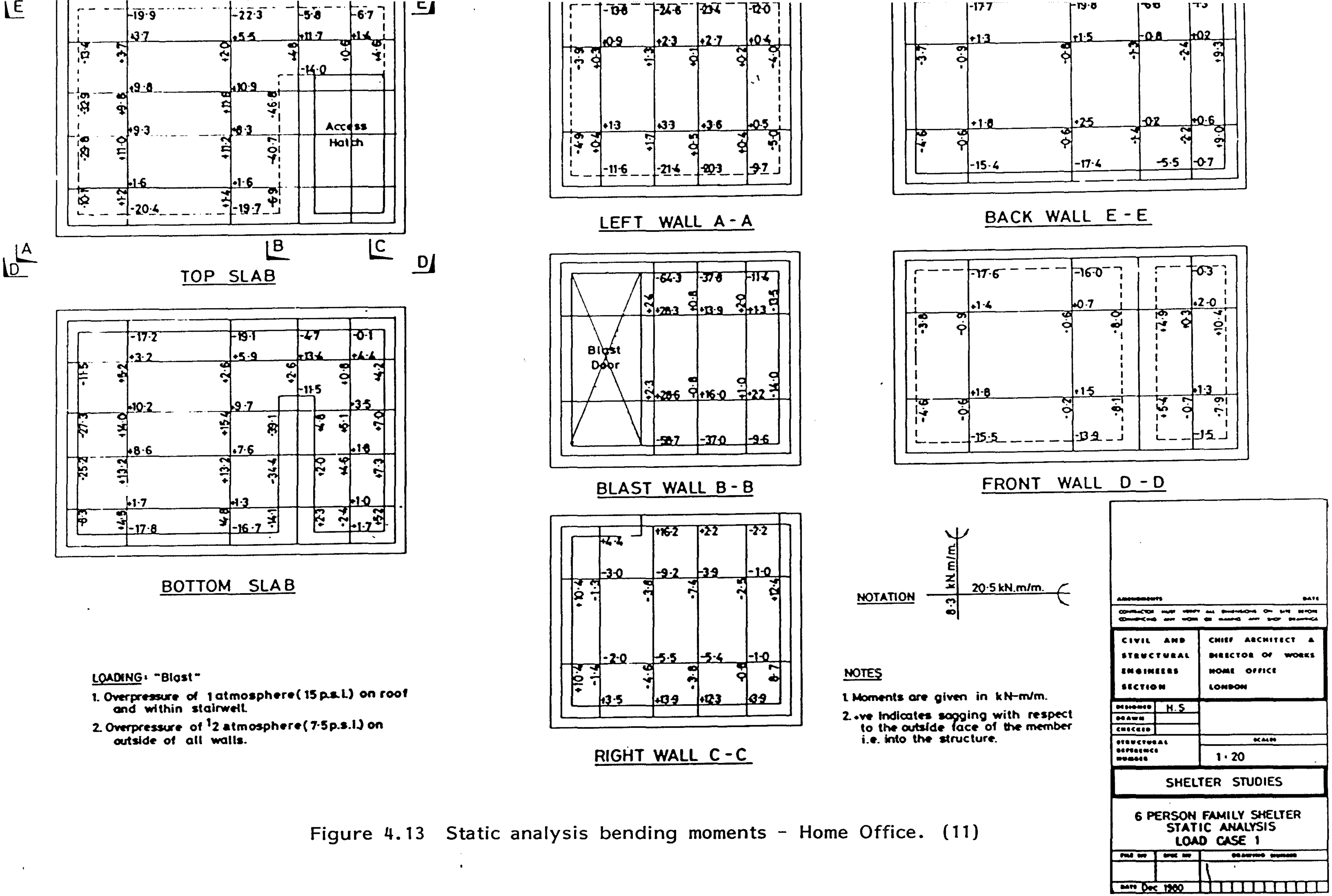


Figure 4.13 Static analysis bending moments - Home Office. (11)

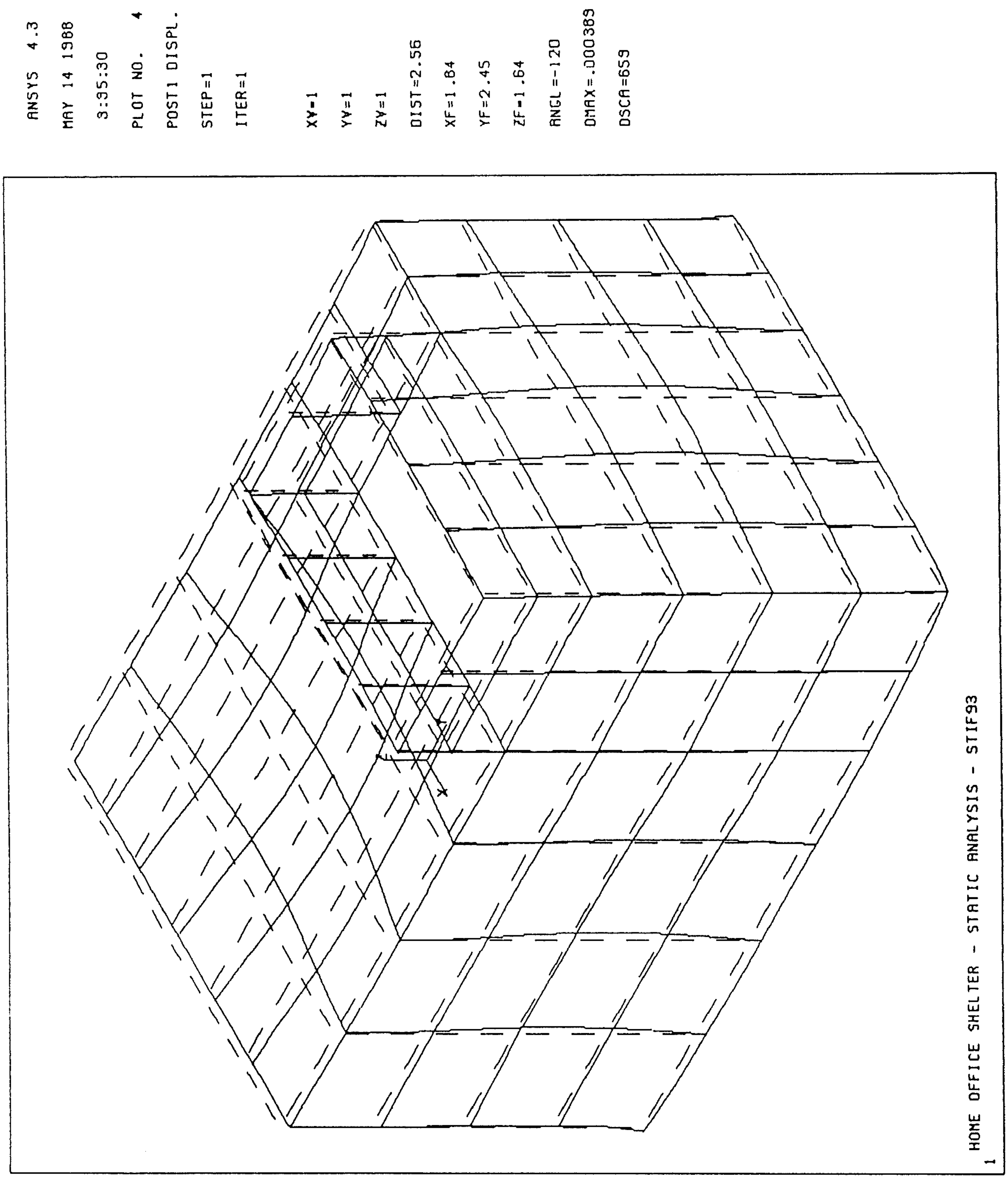


Figure 4.14 Static analysis displacement - Stif93.

Roof and front walls.

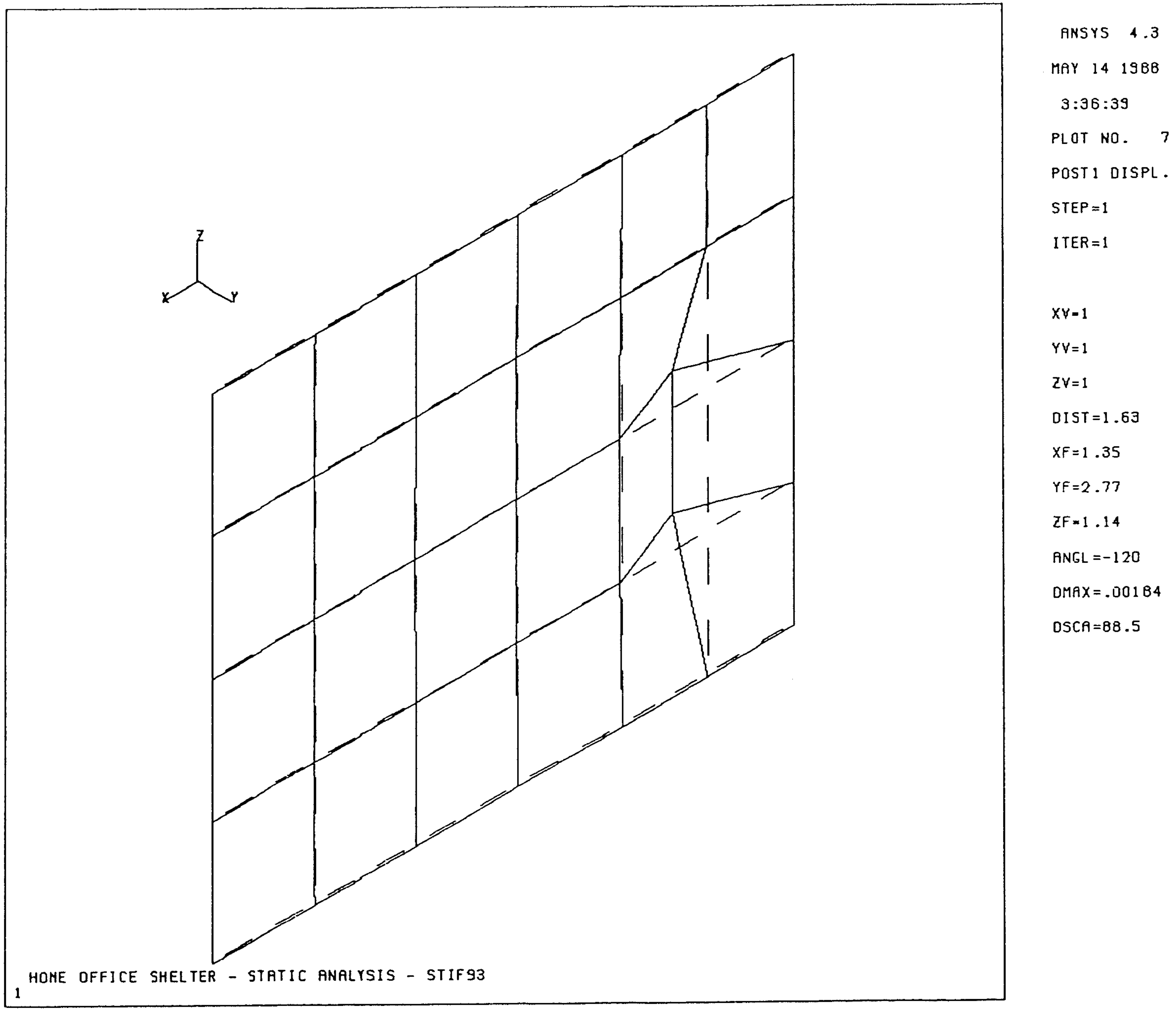


Figure 4.15 Static analysis displacement - Stif93.
Blast wall.

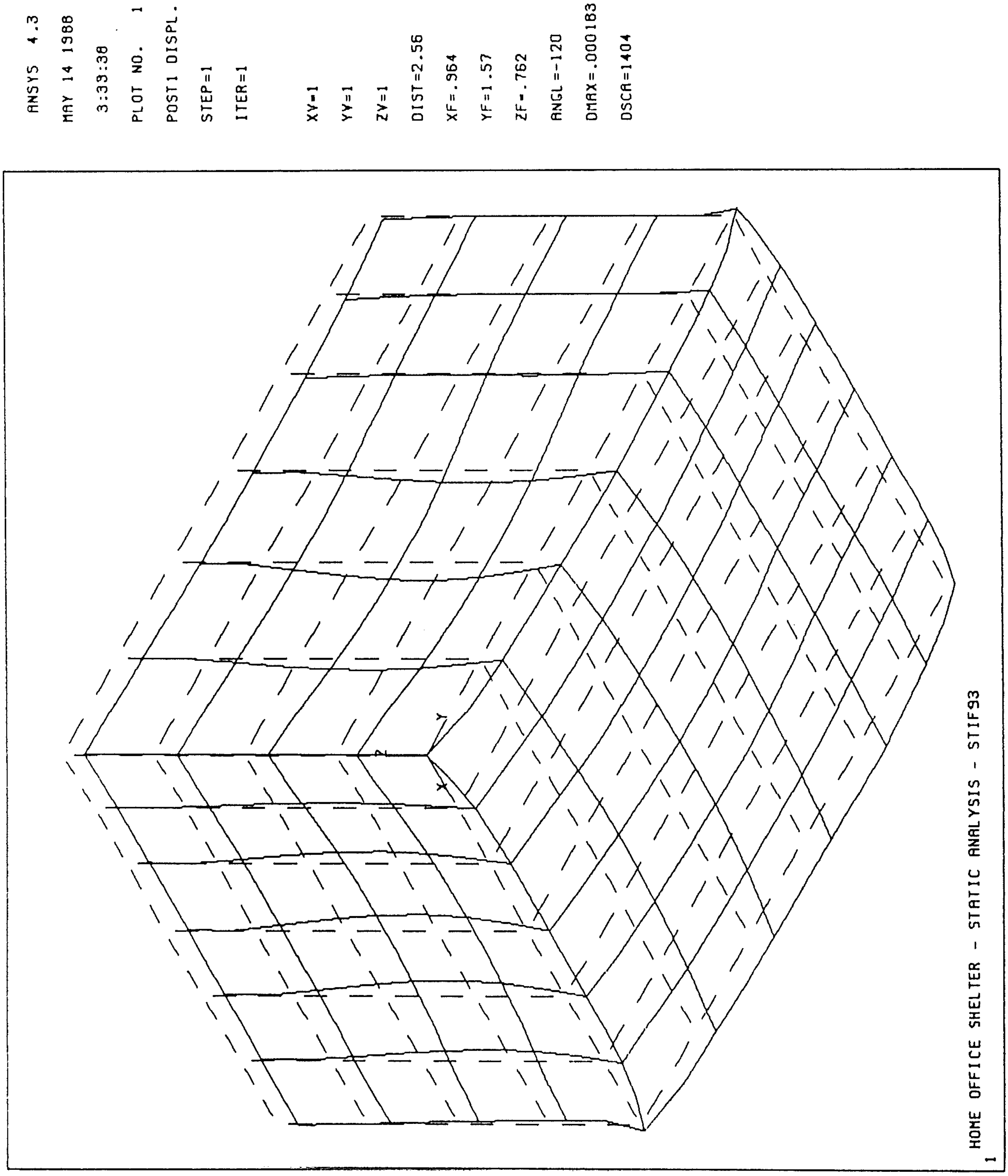


Figure 4.16 Static analysis displacement - Stif93.
Floor and rear walls.

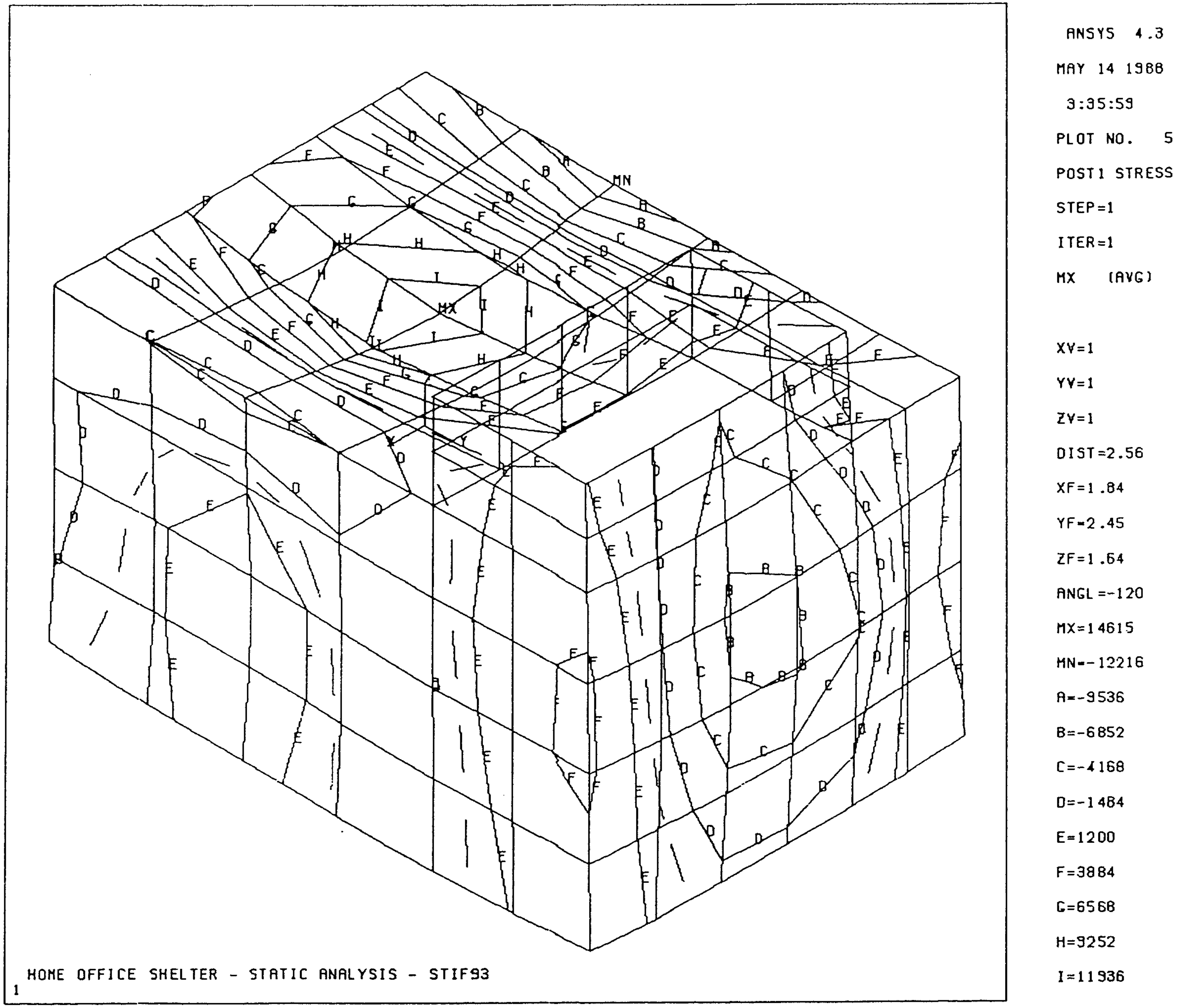


Figure 4.17 Static analysis bending moments - Stif93.
Roof and front walls (MX).

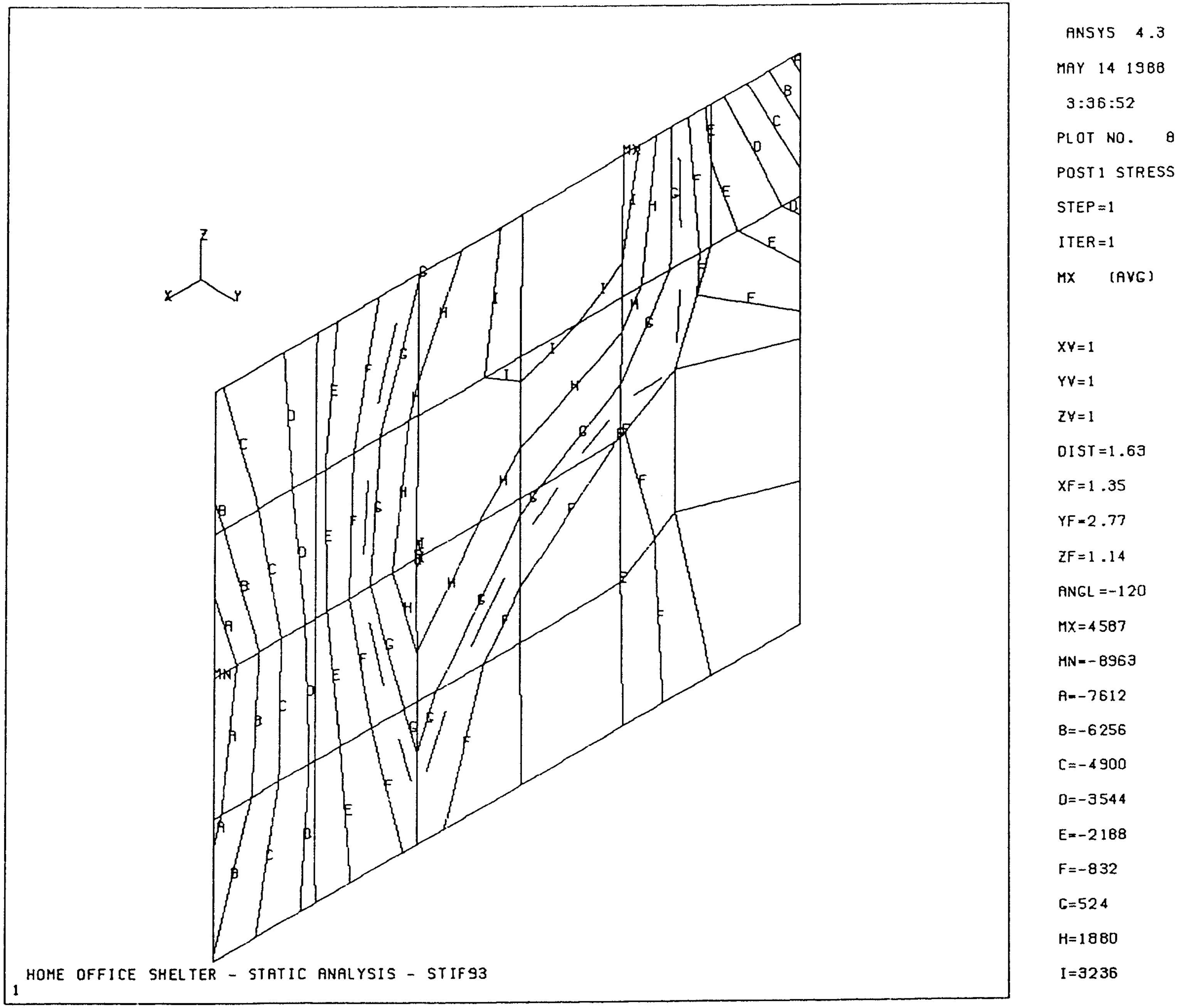


Figure 4.18 Static analysis bending moments - Stif93.
Blast walls (MX).

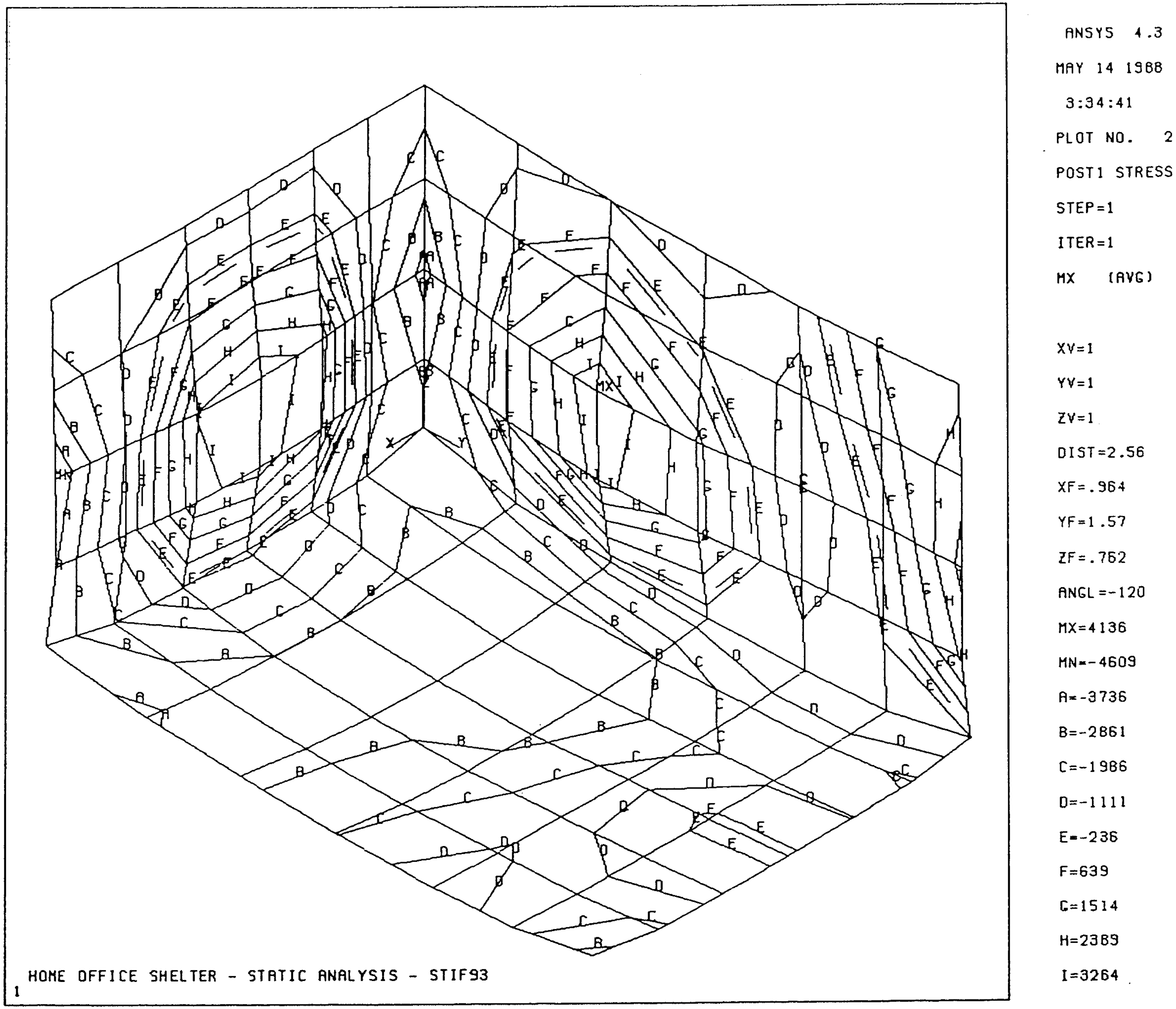


Figure 4.19 Static analysis bending moments - Stif93. Floor and rear walls (MX).

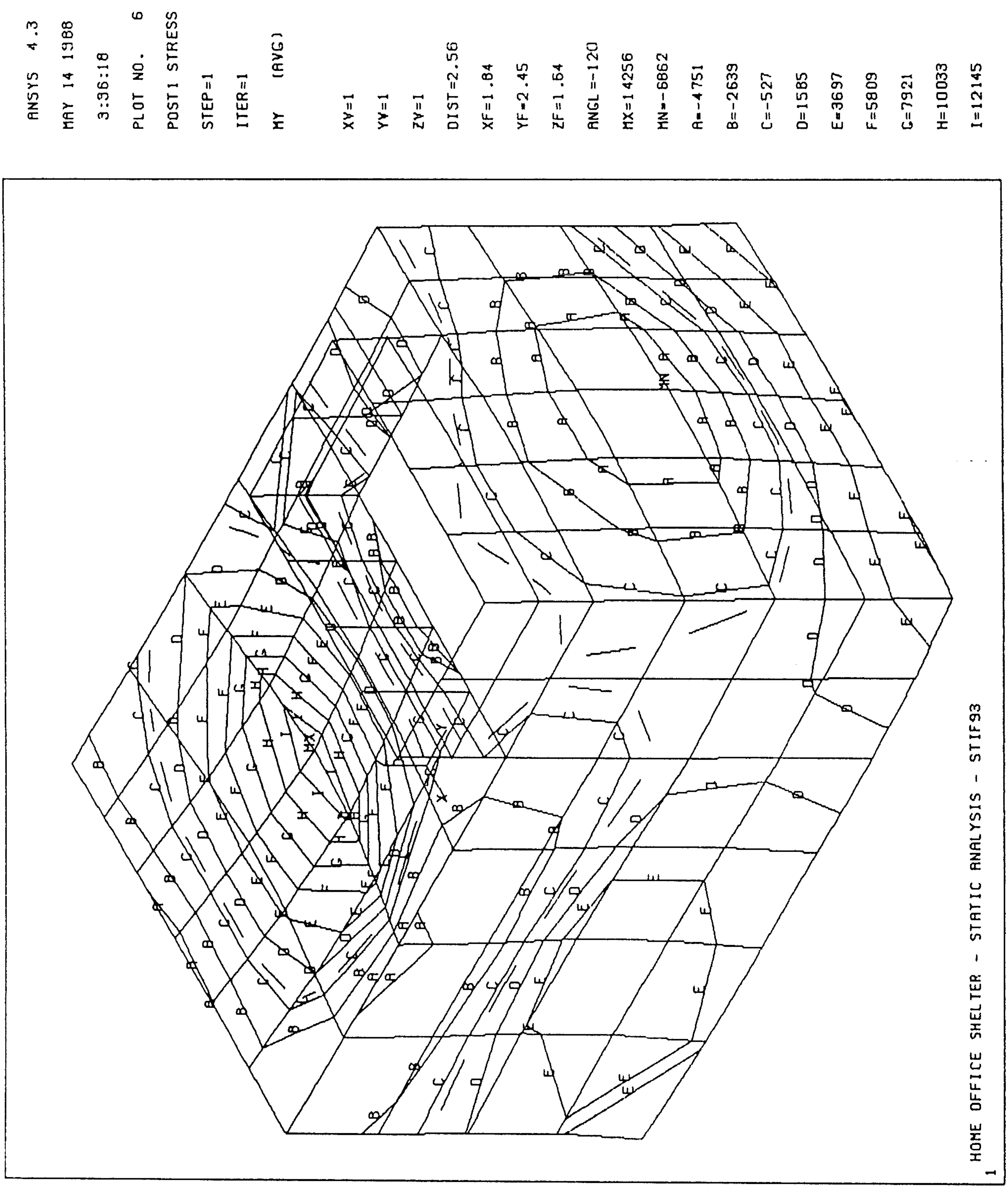


Figure 4.20 Static analysis bending moments - Stif93.
Roof and front walls (MY).

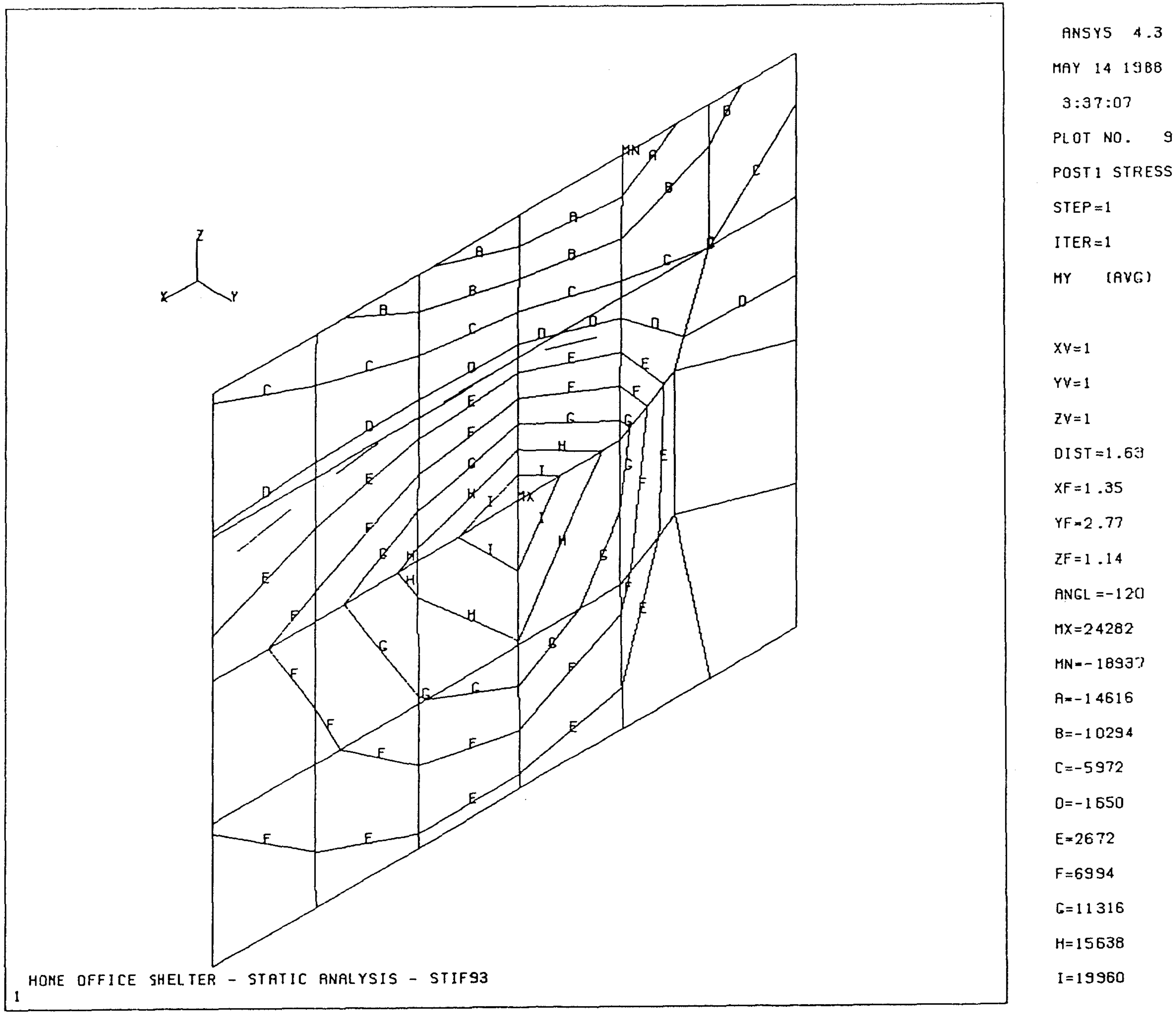


Figure 4.21 Static analysis bending moments - Stif93.
Blast wall (MY)

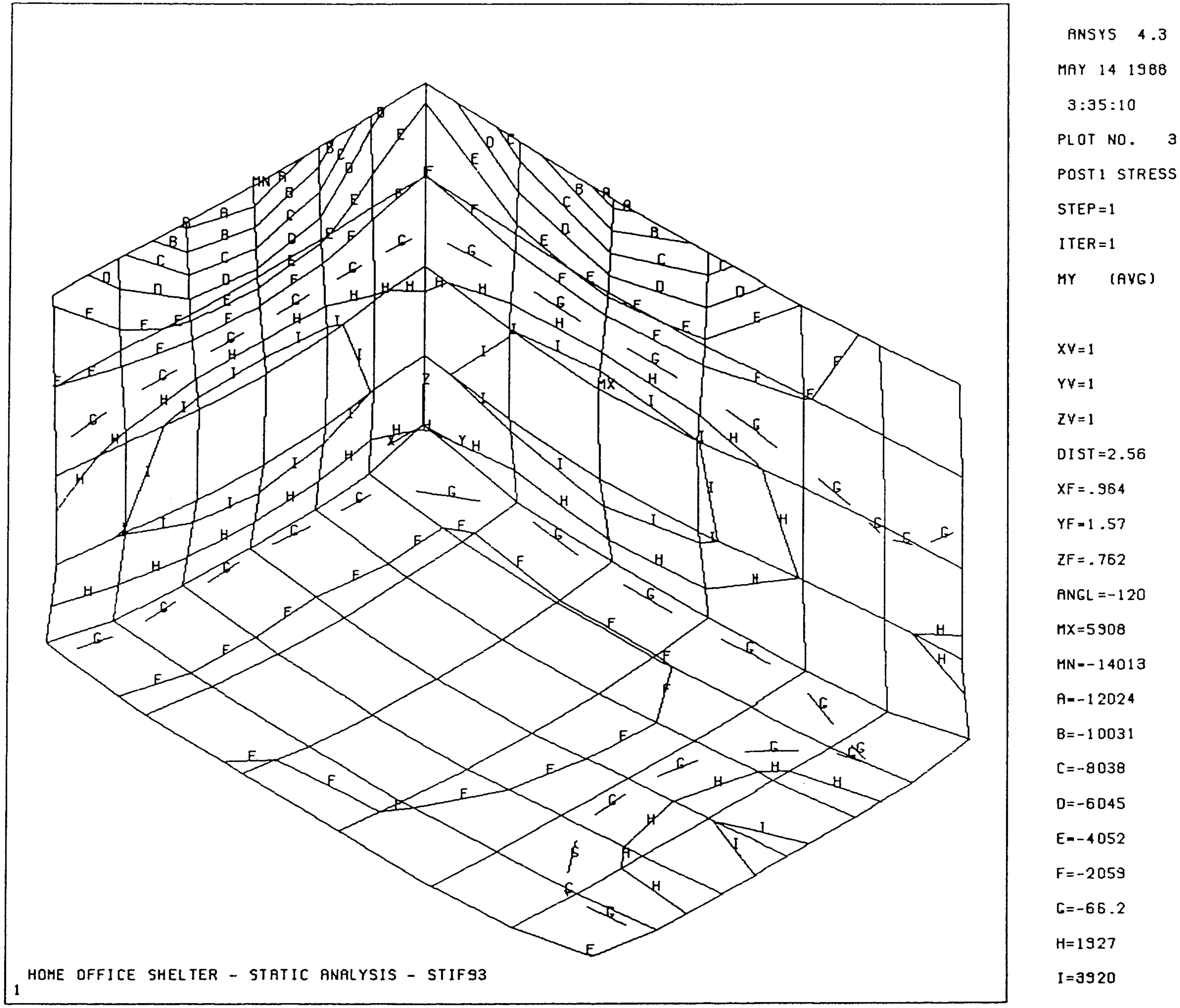


Figure 4.22 Static analysis bending moments - Stif93.
Floor and rear walls (MY)

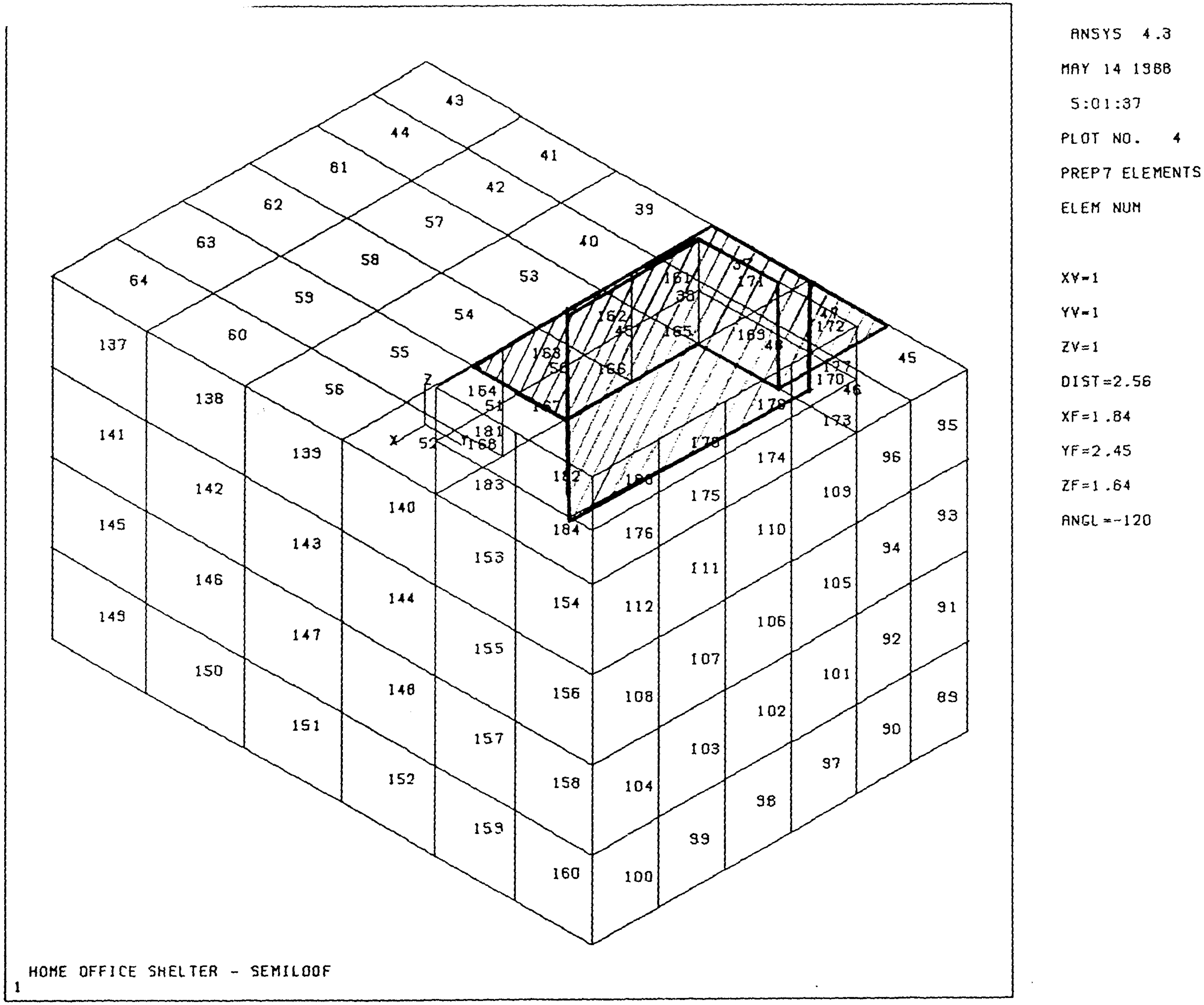


Figure 4.23 Region selected for submodelling. Top of blast wall.

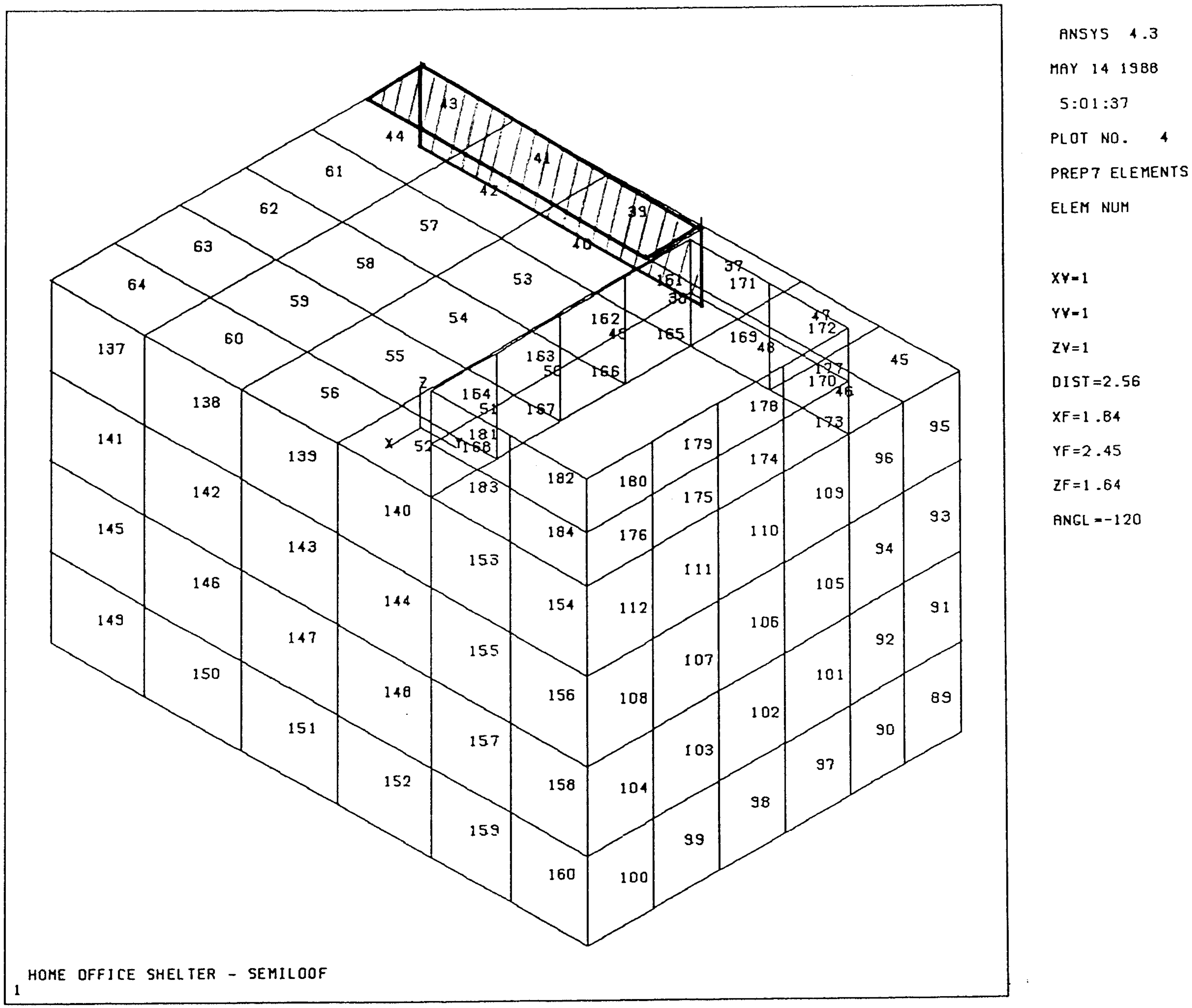


Figure 4.24 Region selected for submodelling.
Top of rear wall.

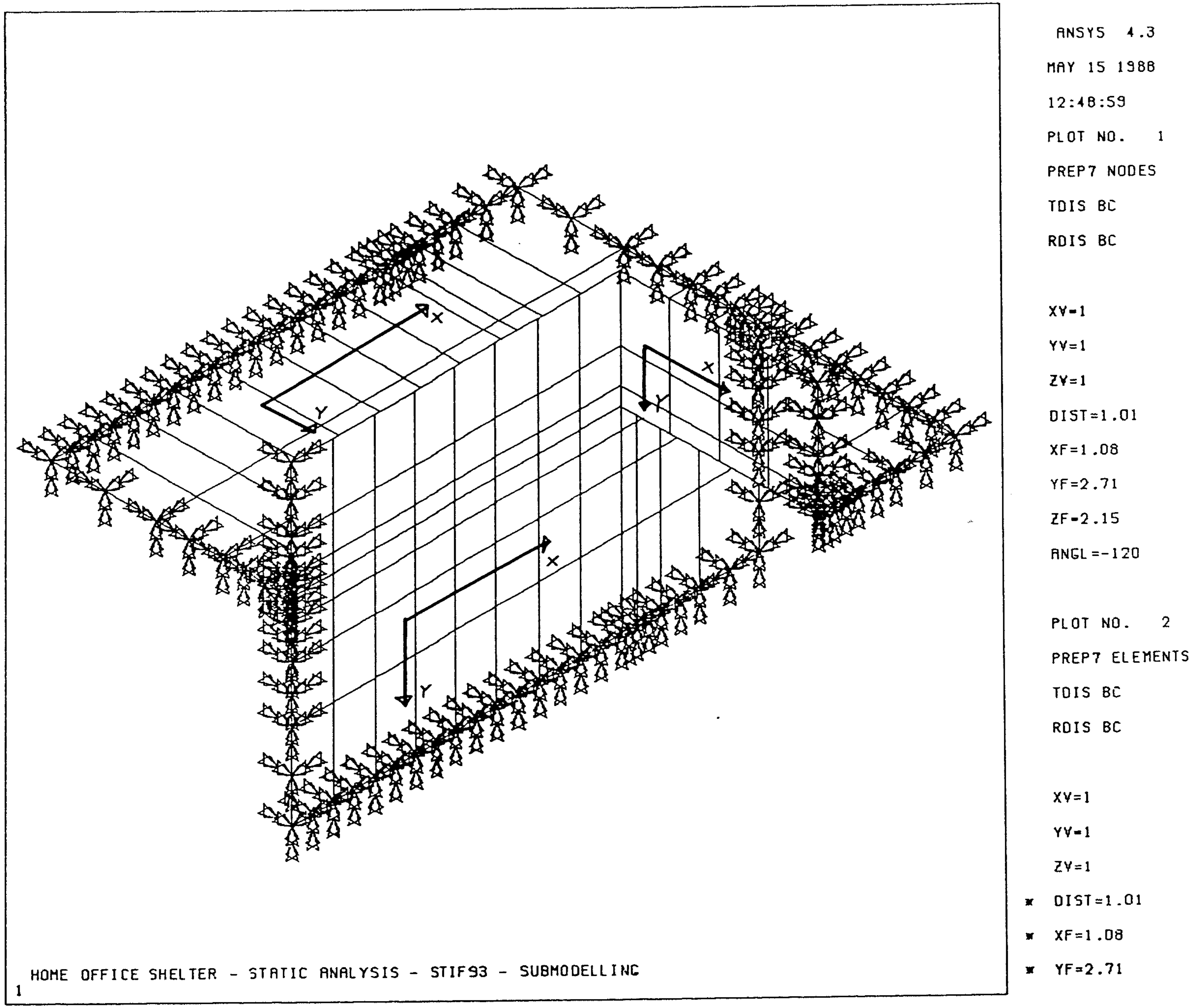


Figure 4.25 Interpolated displacement constraints.
Top of blast wall.

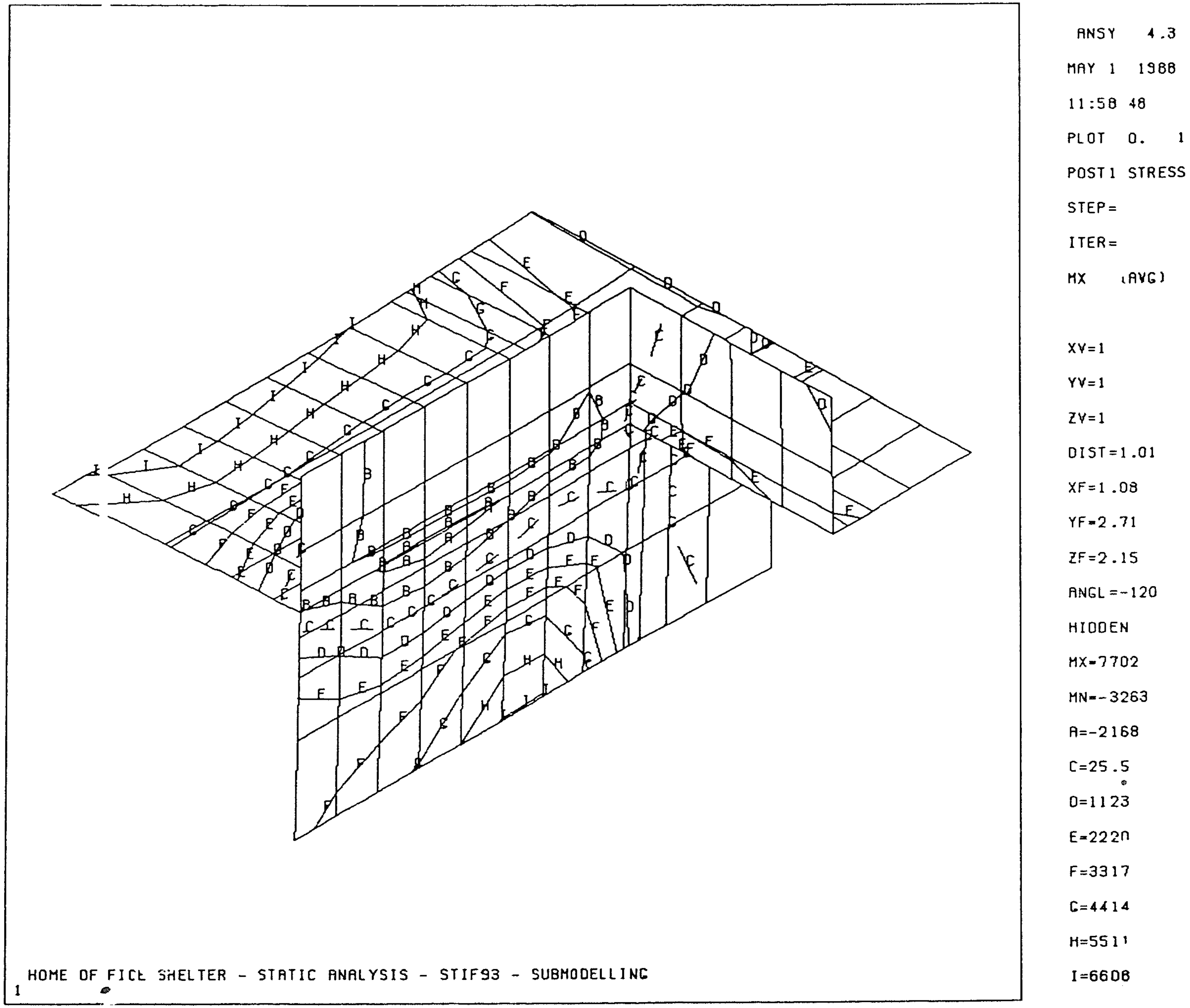


Figure 4.26 Bending moments - Stif93.
Top of blast wall (MX)

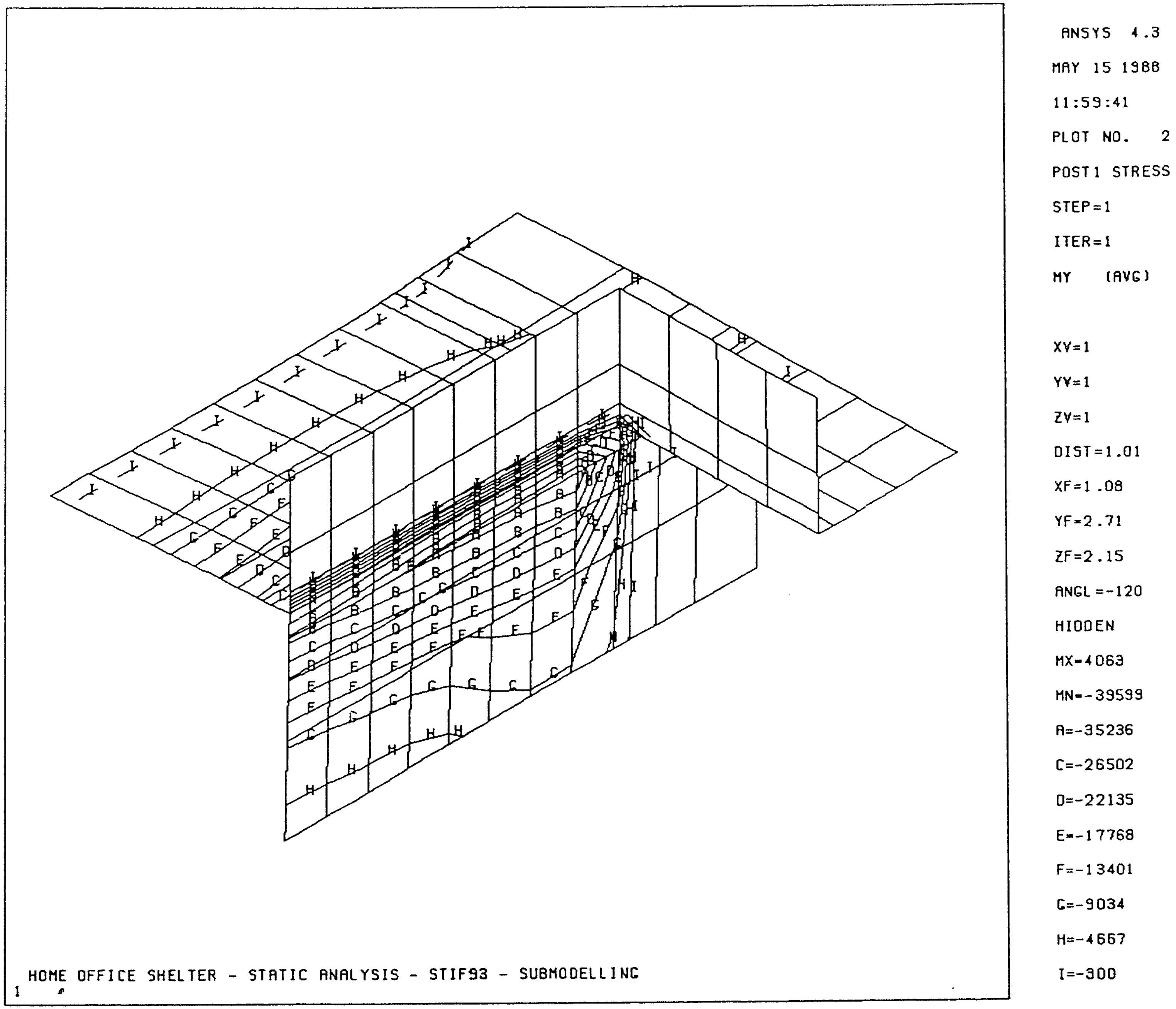


Figure 4.27 Bending moments - Stif93.
Top of blast wall (MY).

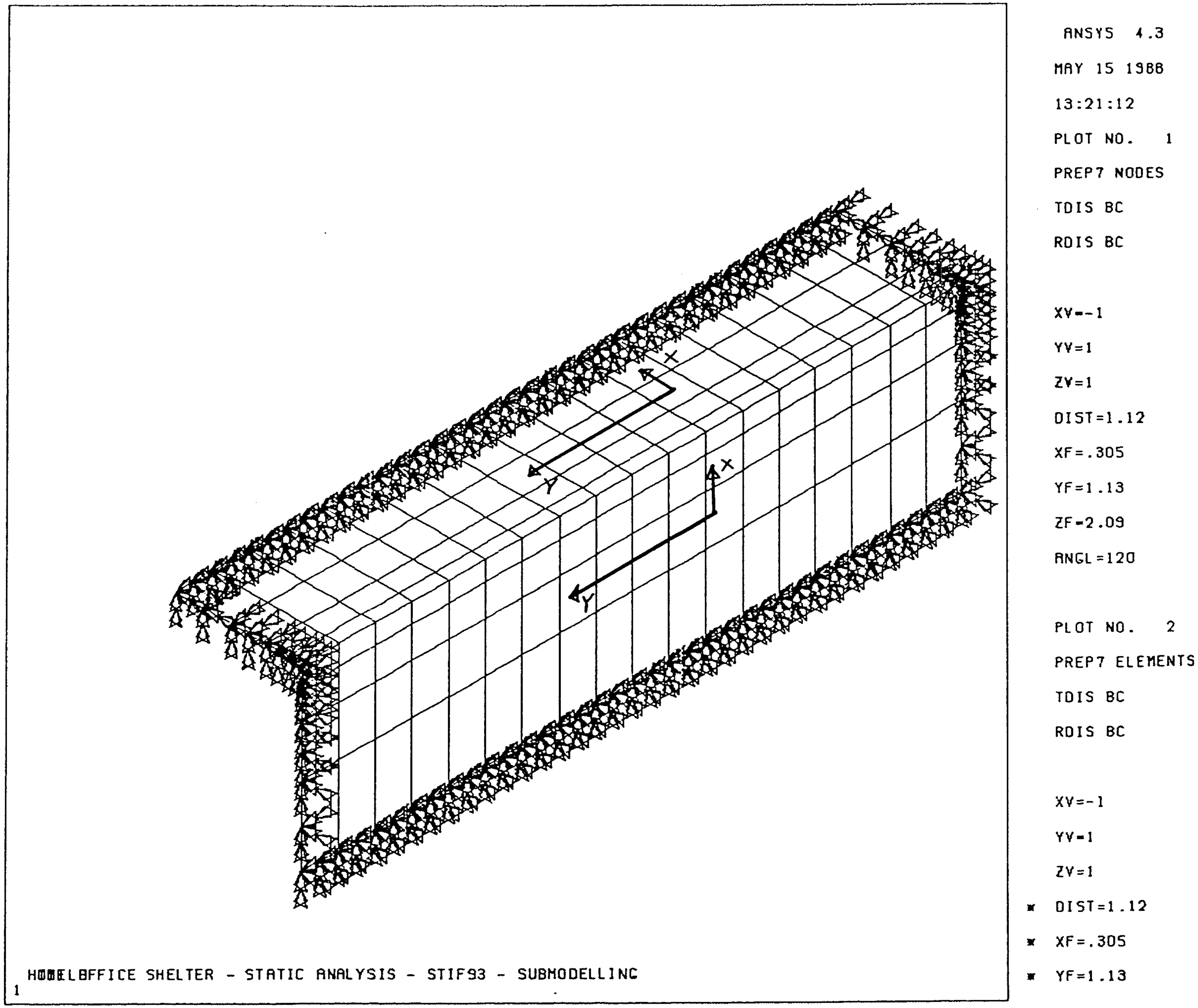


Figure 4.28 Interpolated displacement constraints.
Top of rear wall.

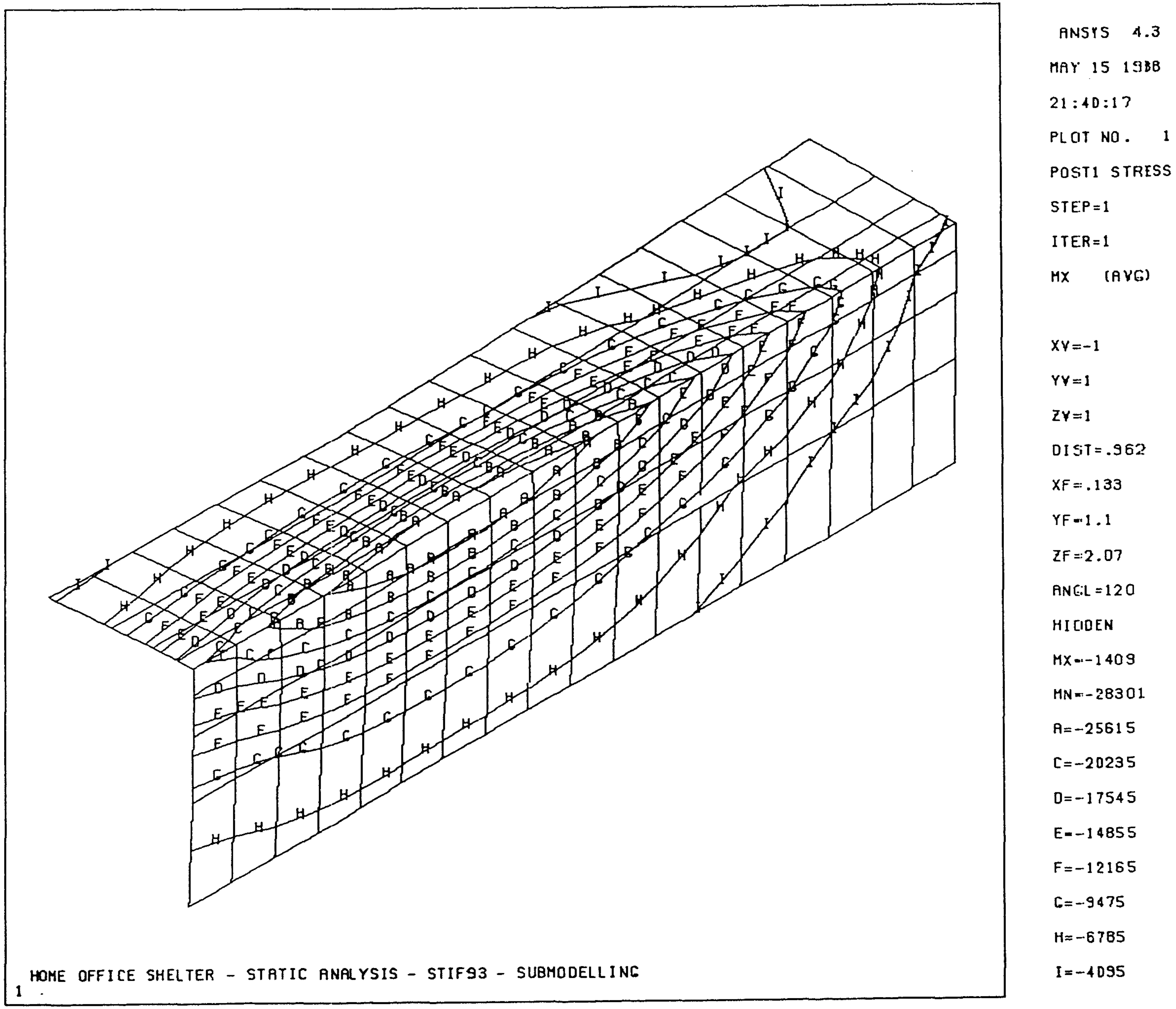


Figure 4.29 Bending moments - Stif93.
Top of rear wall (MX).

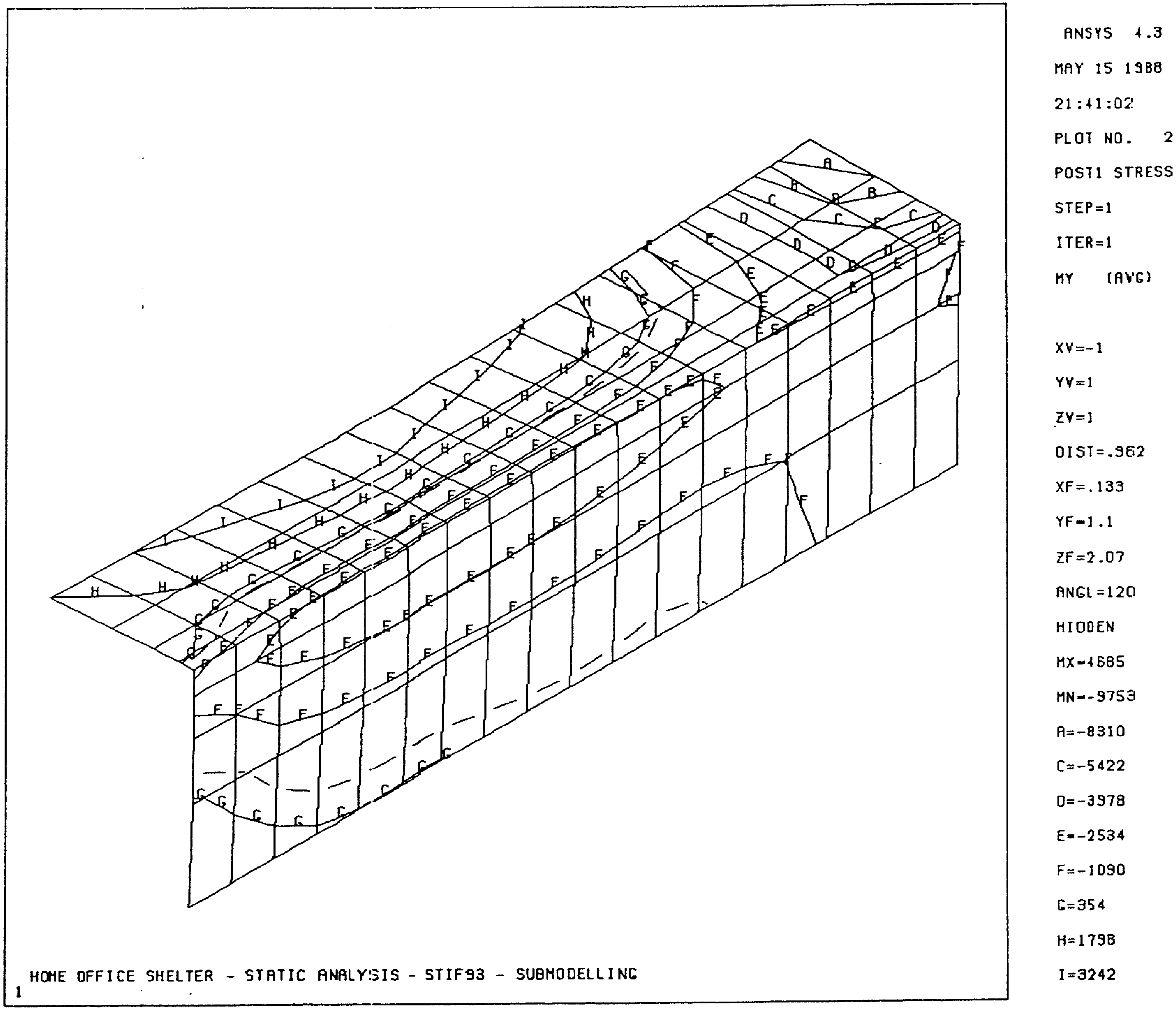


Figure 4.30 Bending moments - Stif93.
Top of rear wall (MY).

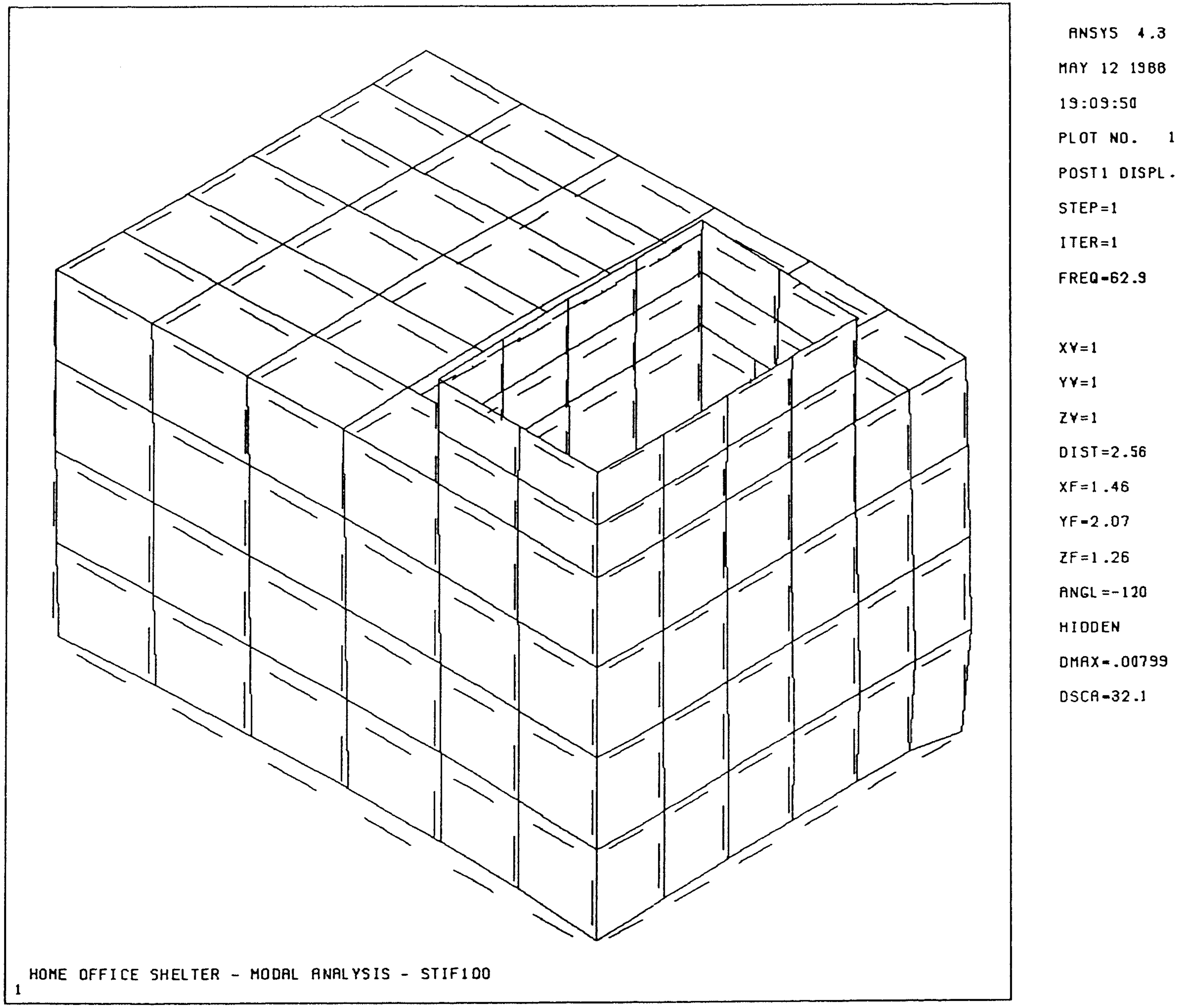


Figure 4.31 Mode shape 1 - SemiLoof.

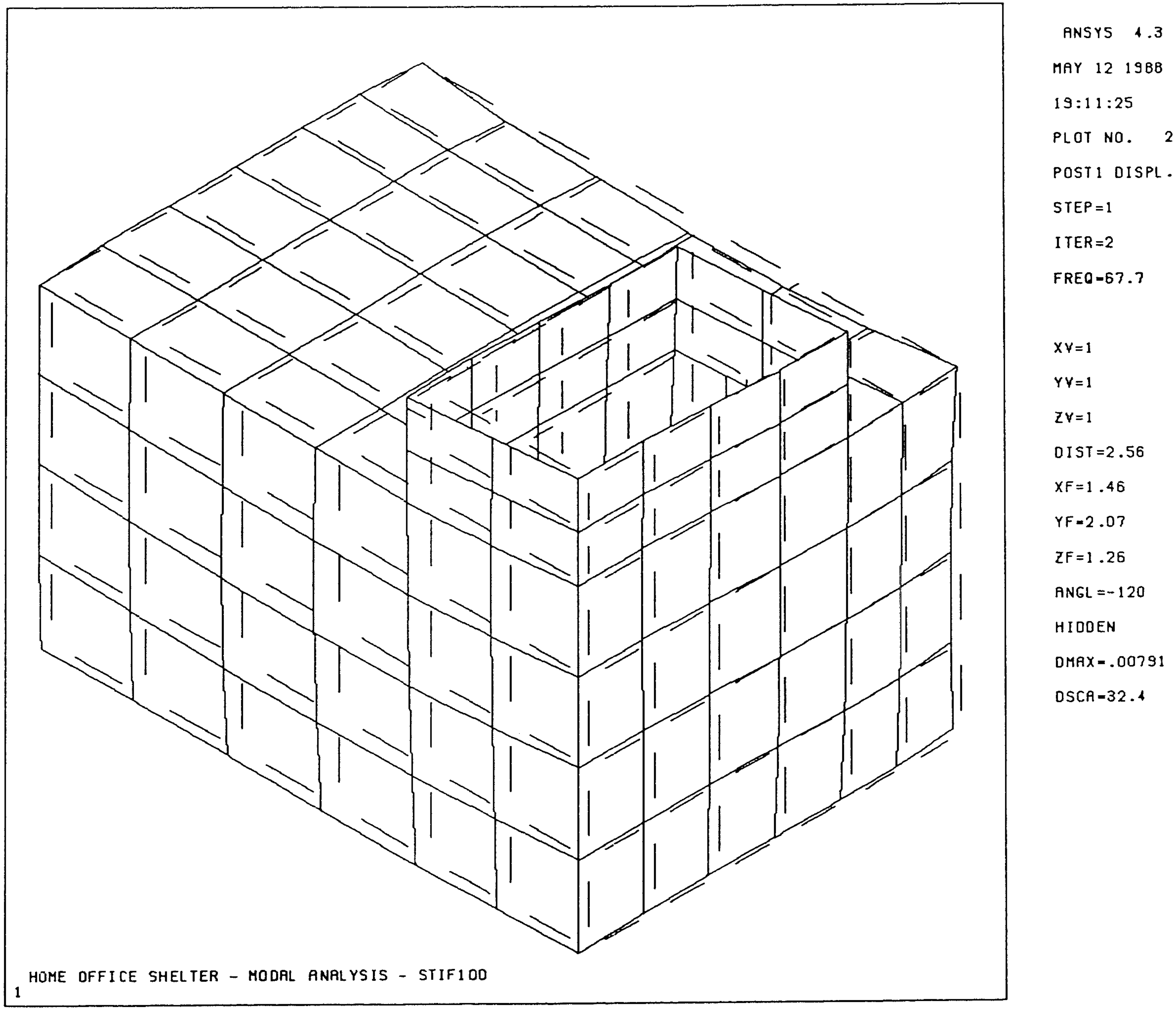


Figure 4.32 Mode shape 2 - SemiLoof.

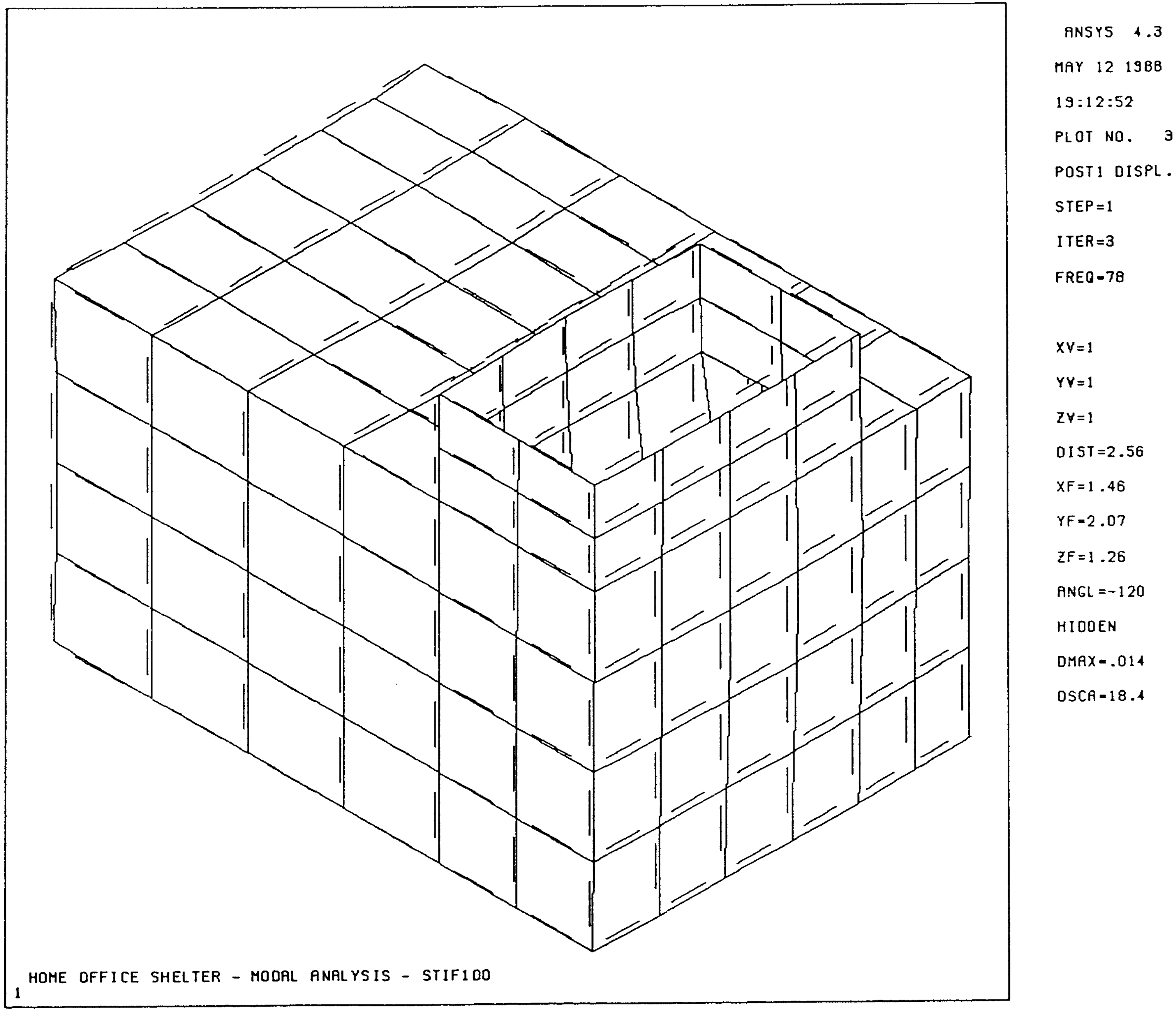


Figure 4.33 Mode shape 3 - SemiLoof.

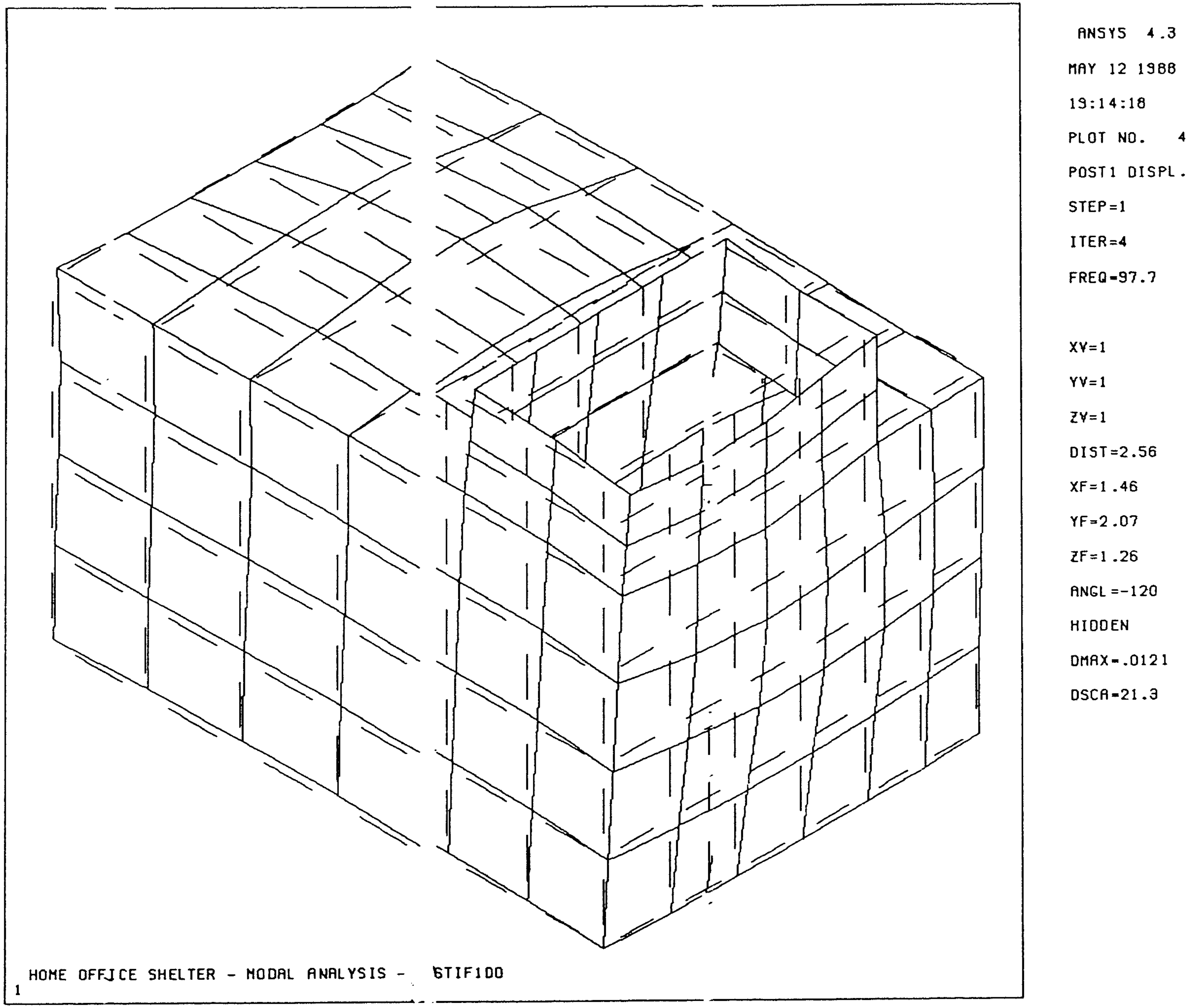


Figure 4.34 Mode shape 4 - SemiLoof.

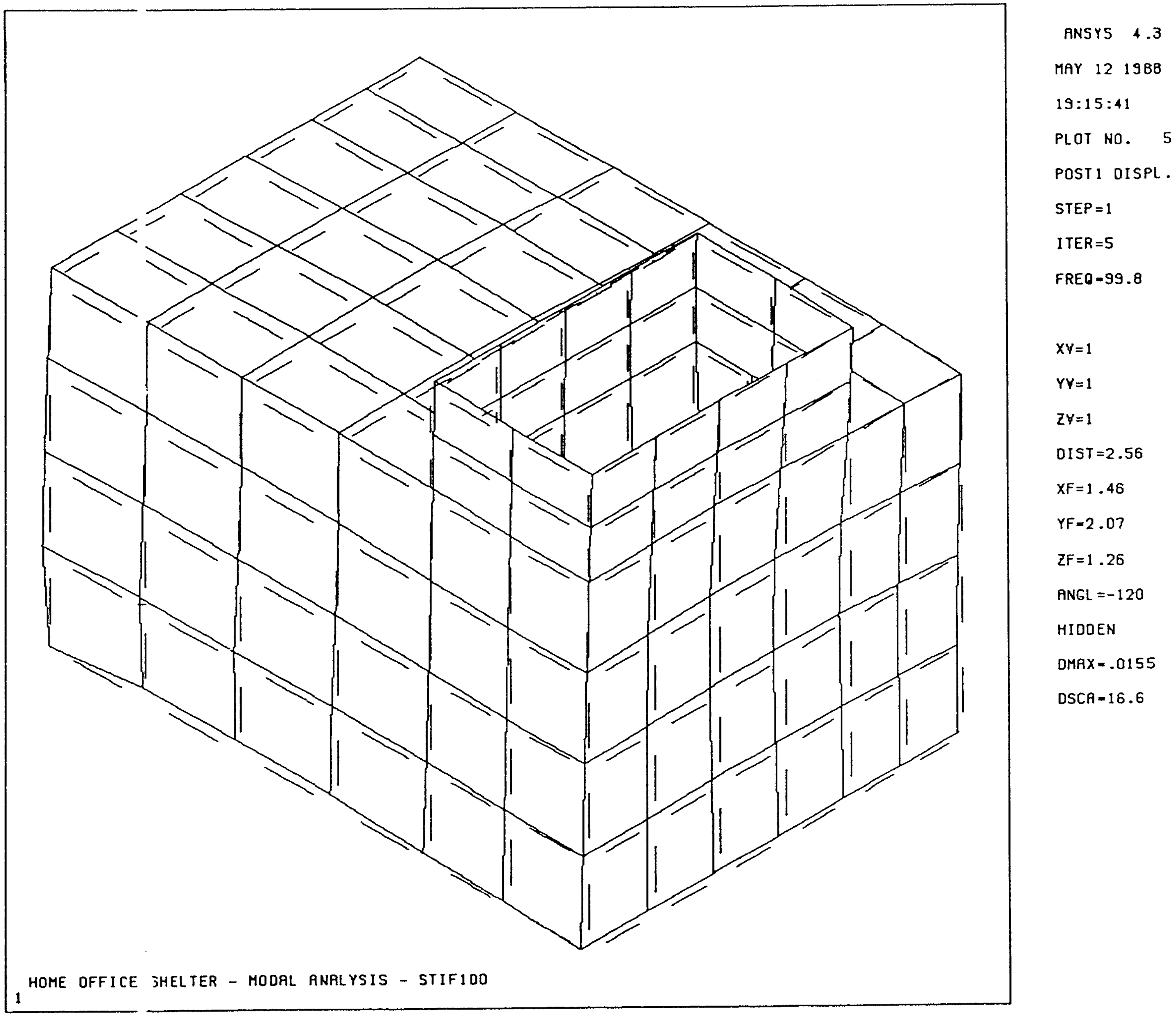


Figure 4.35 Mode shape 5 - SemiLoof.

MASS(X,Y,Z) = 0.3258E+05 0.2159E+05 0.3493E+05 (total masses in X, Y, Z)

***** PARTICIPATION FACTOR CALCULATION ***** X DIRECTION

MODE	FREQUENCY	PERIOD	PARTIC.FACTOR	RATIO	EFFECTIVE MASS	CUMULATIVE MASS FRACTION
1	62.8582	0.15909E-01	54.734	0.354843	2995.83	0.102578
2	67.6664	0.14778E-01	154.25	1.000000	23792.8	0.917245
3	78.0386	0.12814E-01	-2.4463	0.015859	5.98422	0.917450
4	97.6707	0.10238E-01	-5.7796	0.037470	33.4043	0.918594
5	99.8073	0.10019E-01	-6.2255	0.040360	38.7566	0.919921
6	107.134	0.93341E-02	-37.659	0.244141	1418.17	0.968479
7	108.893	0.91833E-02	7.5916	0.049216	57.6323	0.970453
8	118.219	0.84589E-02	-10.521	0.068209	110.694	0.974243
9	119.714	0.83532E-02	-27.162	0.176094	737.788	0.999505
10	123.824	0.80760E-02	3.8028	0.024654	14.4615	1.00000
			SUM OF EFFECTIVE MASSES=		29205.5	

***** PARTICIPATION FACTOR CALCULATION ***** Y DIRECTION

MODE	FREQUENCY	PERIOD	PARTIC.FACTOR	RATIO	EFFECTIVE MASS	CUMULATIVE MASS FRACTION
1	62.8582	0.15909E-01	3.2074	0.027535	10.2871	0.535928E-03
2	67.6664	0.14778E-01	-9.6029	0.082441	92.2153	0.534007E-02
3	78.0386	0.12814E-01	-116.48	1.000000	13567.9	0.712188
4	97.6707	0.10238E-01	60.546	0.519794	3665.86	0.903168
5	99.8073	0.10019E-01	-36.154	0.310381	1307.09	0.971263
6	107.134	0.93341E-02	-15.027	0.129007	225.809	0.983027
7	108.893	0.91833E-02	-4.7144	0.040473	22.2252	0.984185
8	118.219	0.84589E-02	8.9550	0.076879	80.1912	0.988363
9	119.714	0.83532E-02	-14.878	0.127729	221.358	0.999895
10	123.824	0.80760E-02	1.4213	0.012202	2.02000	1.00000
			SUM OF EFFECTIVE MASSES=		19195.0	

***** PARTICIPATION FACTOR CALCULATION ***** Z DIRECTION

MODE	FREQUENCY	PERIOD	PARTIC.FACTOR	RATIO	EFFECTIVE MASS	CUMULATIVE MASS FRACTION
1	62.8582	0.15909E-01	-146.83	1.000000	21557.7	0.709339
2	67.6664	0.14778E-01	51.824	0.352962	2685.70	0.797710
3	78.0386	0.12814E-01	-0.36382	0.002478	0.132365	0.797714
4	97.6707	0.10238E-01	32.915	0.224176	1083.38	0.833362
5	99.8073	0.10019E-01	54.389	0.370432	2958.14	0.930697
6	107.134	0.93341E-02	21.034	0.143257	442.419	0.945255
7	108.893	0.91833E-02	15.186	0.103431	230.625	0.952843
8	118.219	0.84589E-02	23.390	0.159306	547.100	0.970845
9	119.714	0.83532E-02	-20.989	0.142952	440.535	0.985340
10	123.824	0.80760E-02	21.107	0.143758	445.521	1.00000
			SUM OF EFFECTIVE MASSES=		30391.2	

Table 4.1 Effective modal masses using SemiLoof.

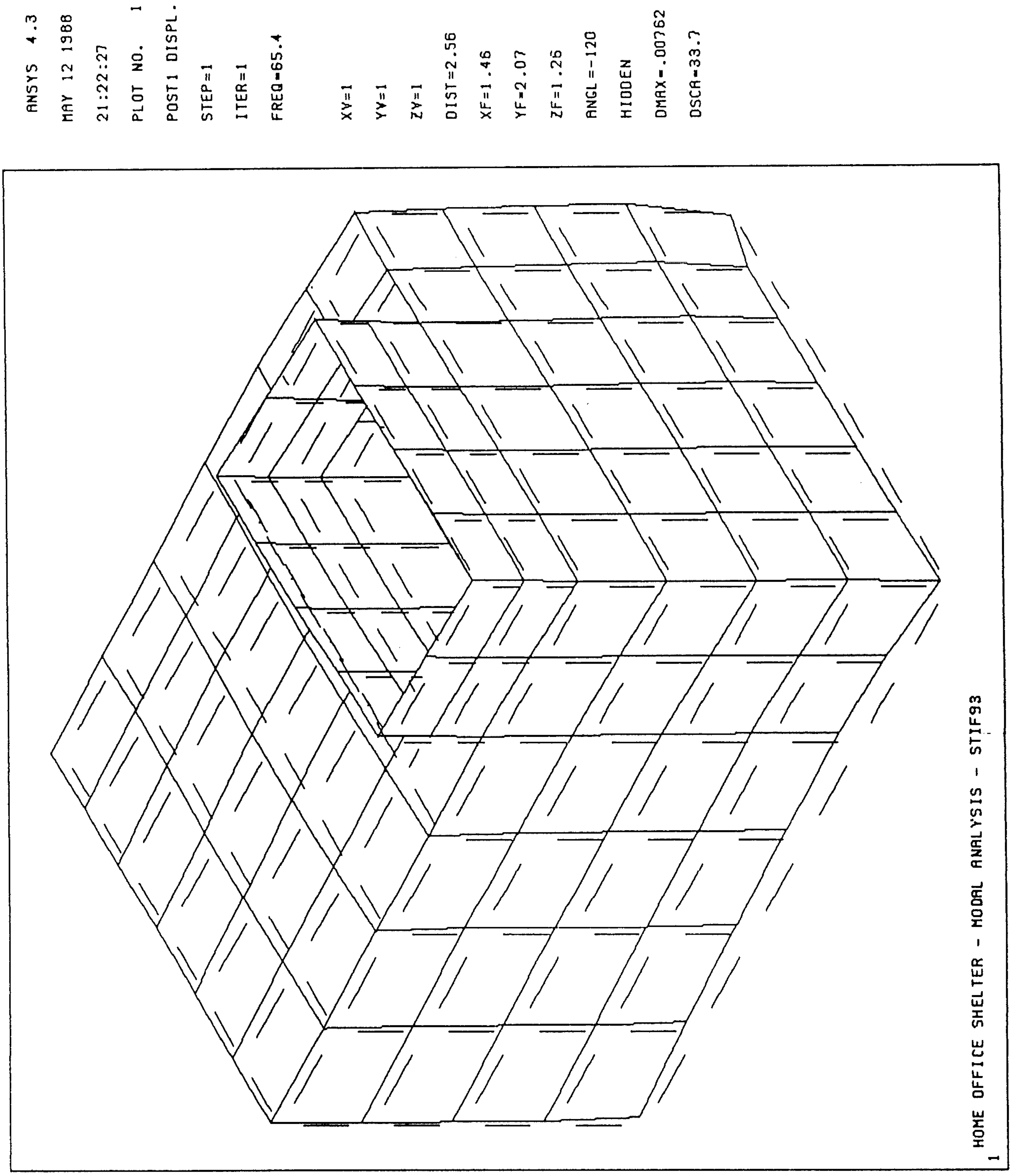


Figure 4.36 Mode shape 1 - Stif93.

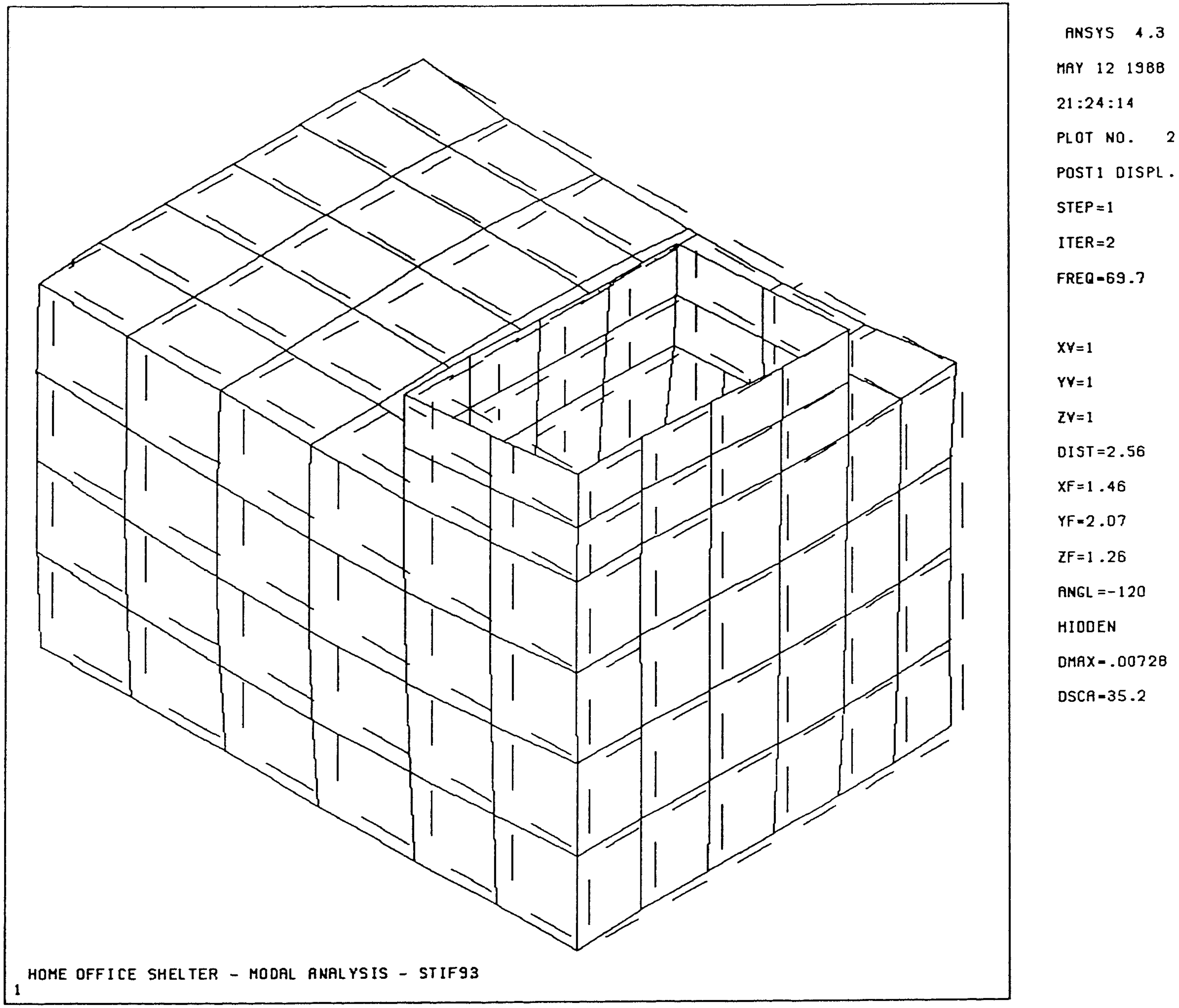


Figure 4.37 Mode shape 2 - Stif93.

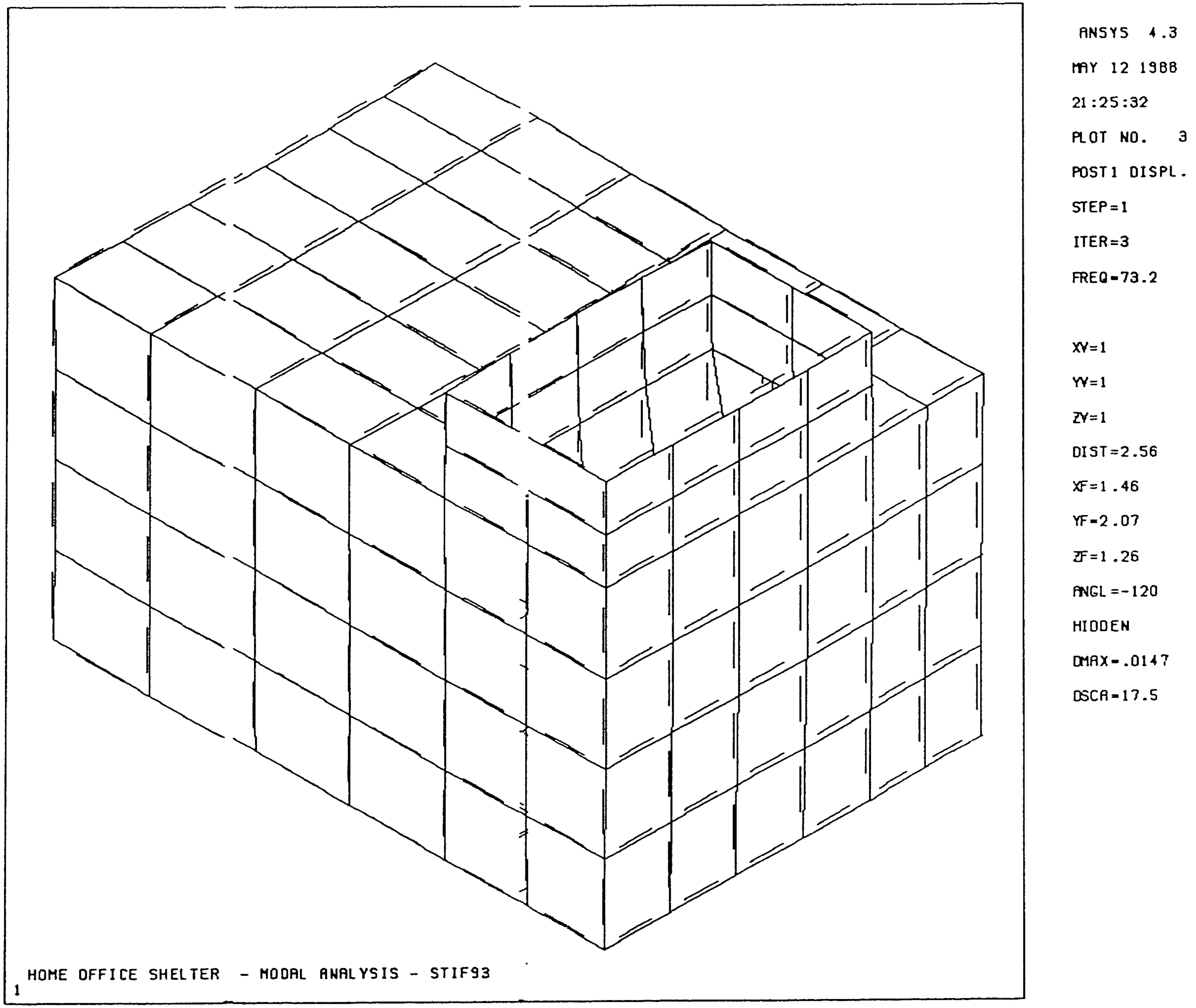


Figure 4.38 Mode shape 3 - Stif93.

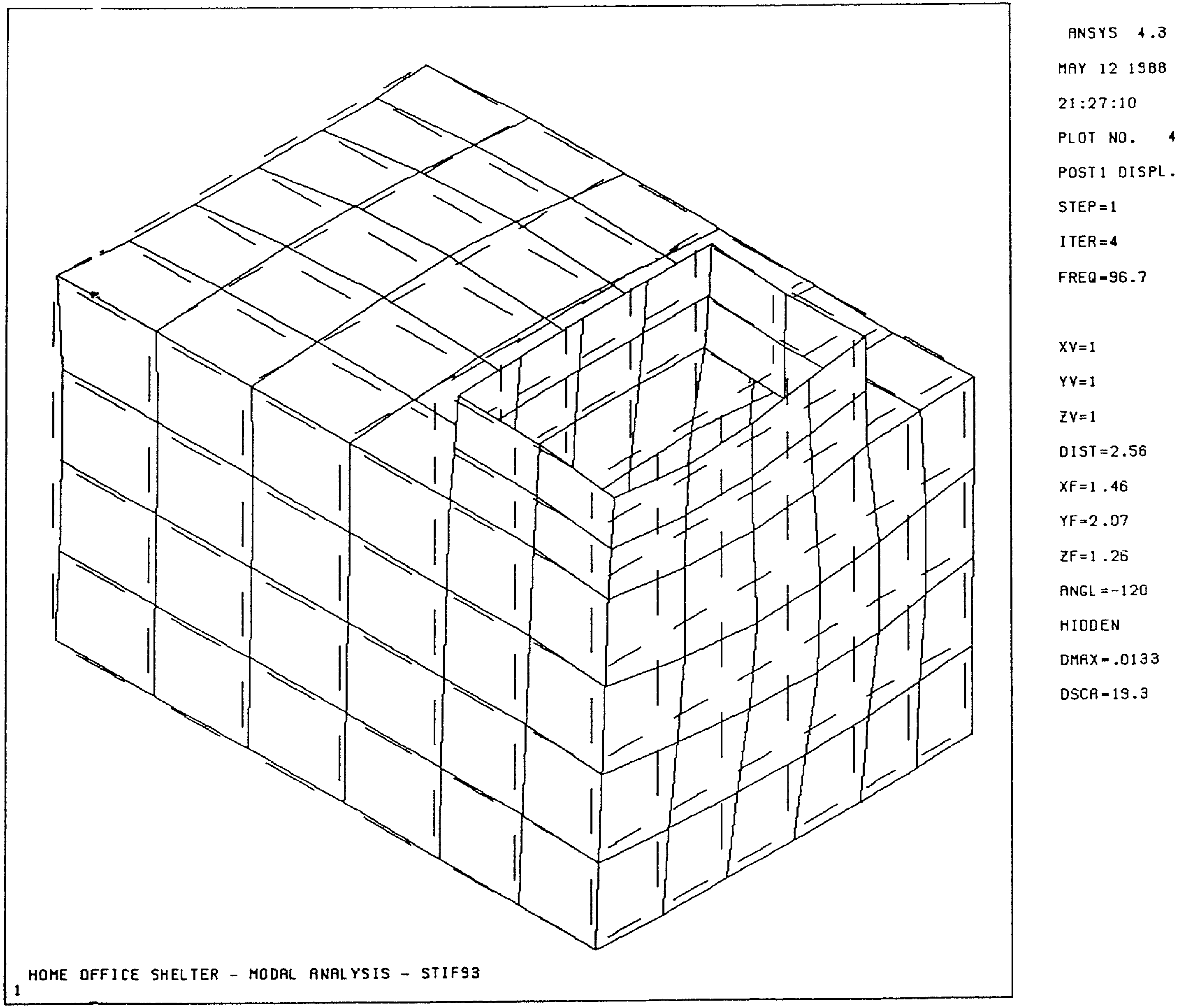


Figure 4.39 Mode shape 4 - Stif93.

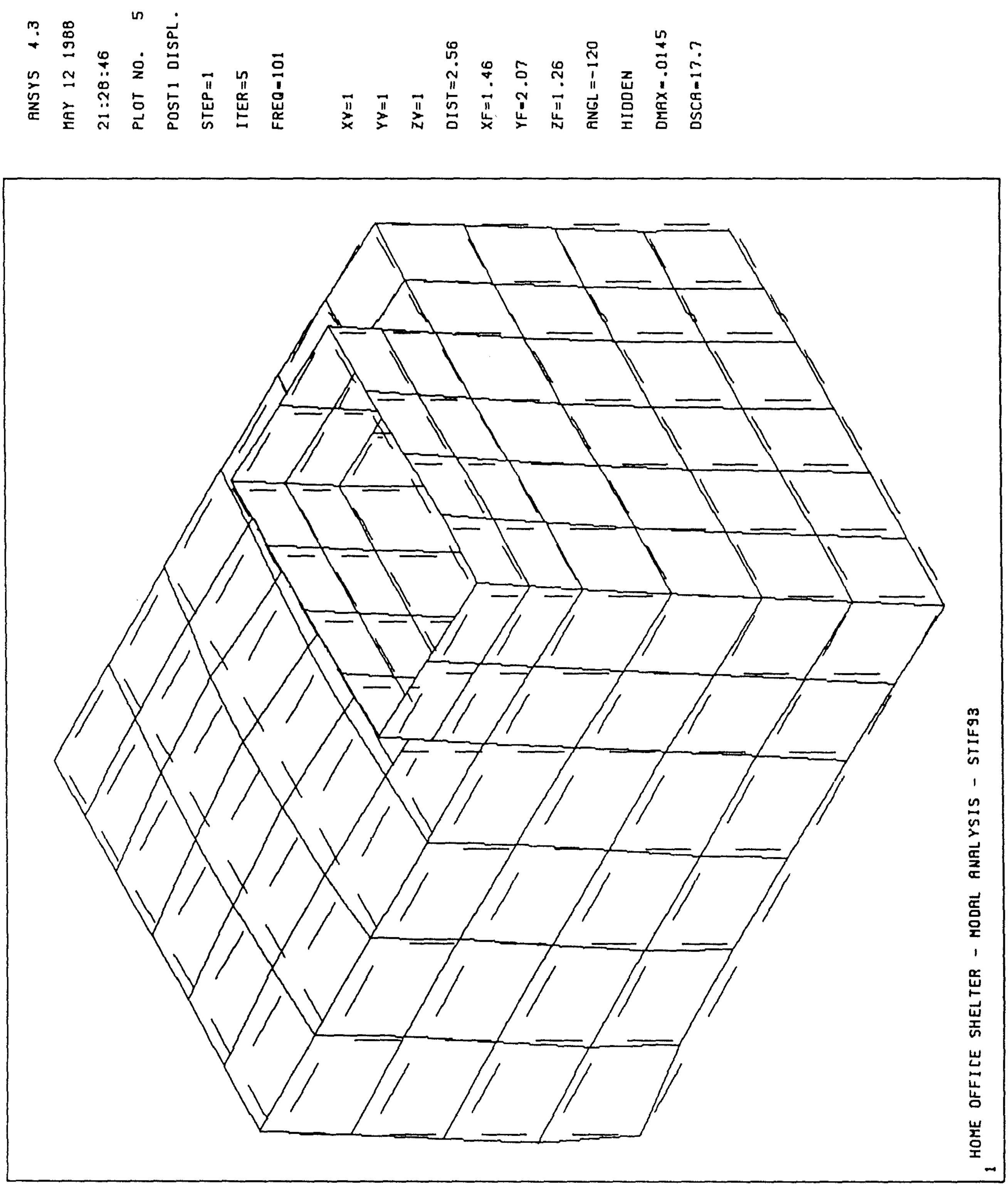
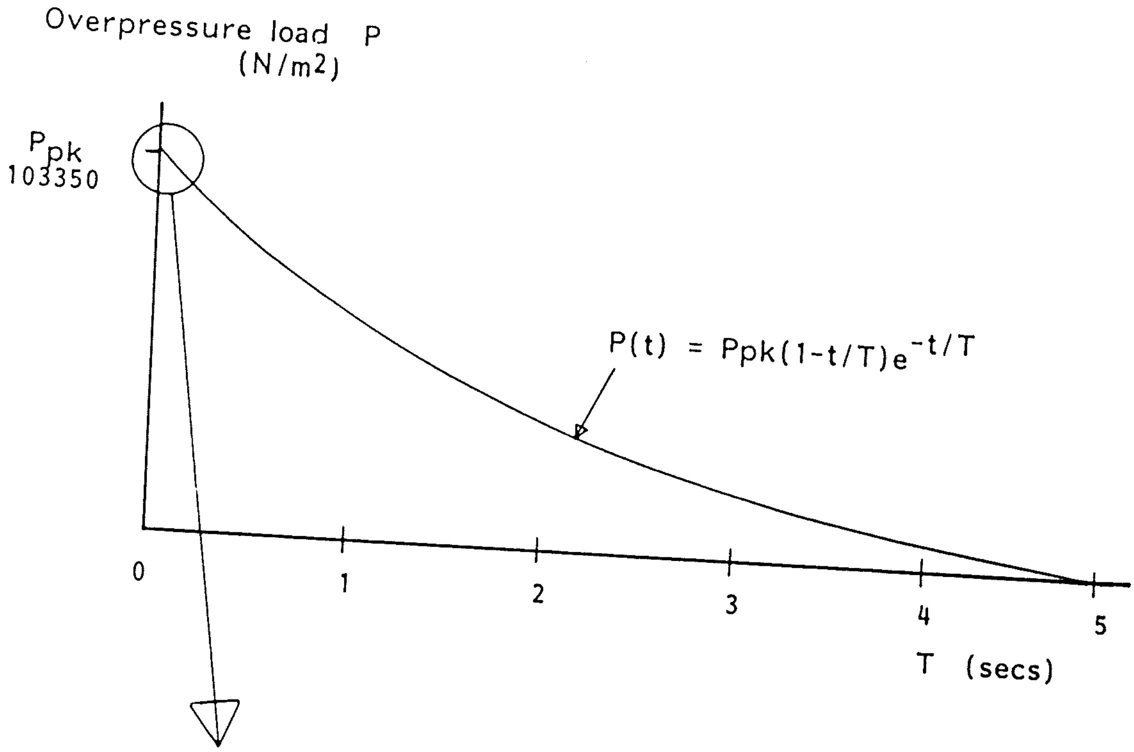


Figure 4.40 Mode shape 5 - Stif93.



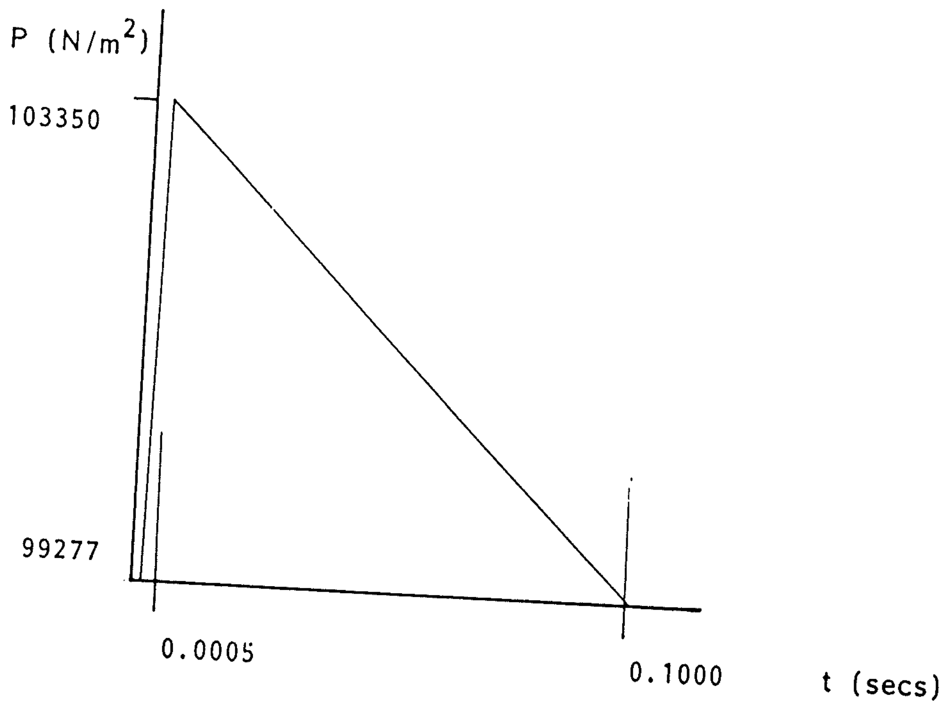


Figure 4.41 Overpressure versus time loading curve for 10MT burst at 7000m from ground zero

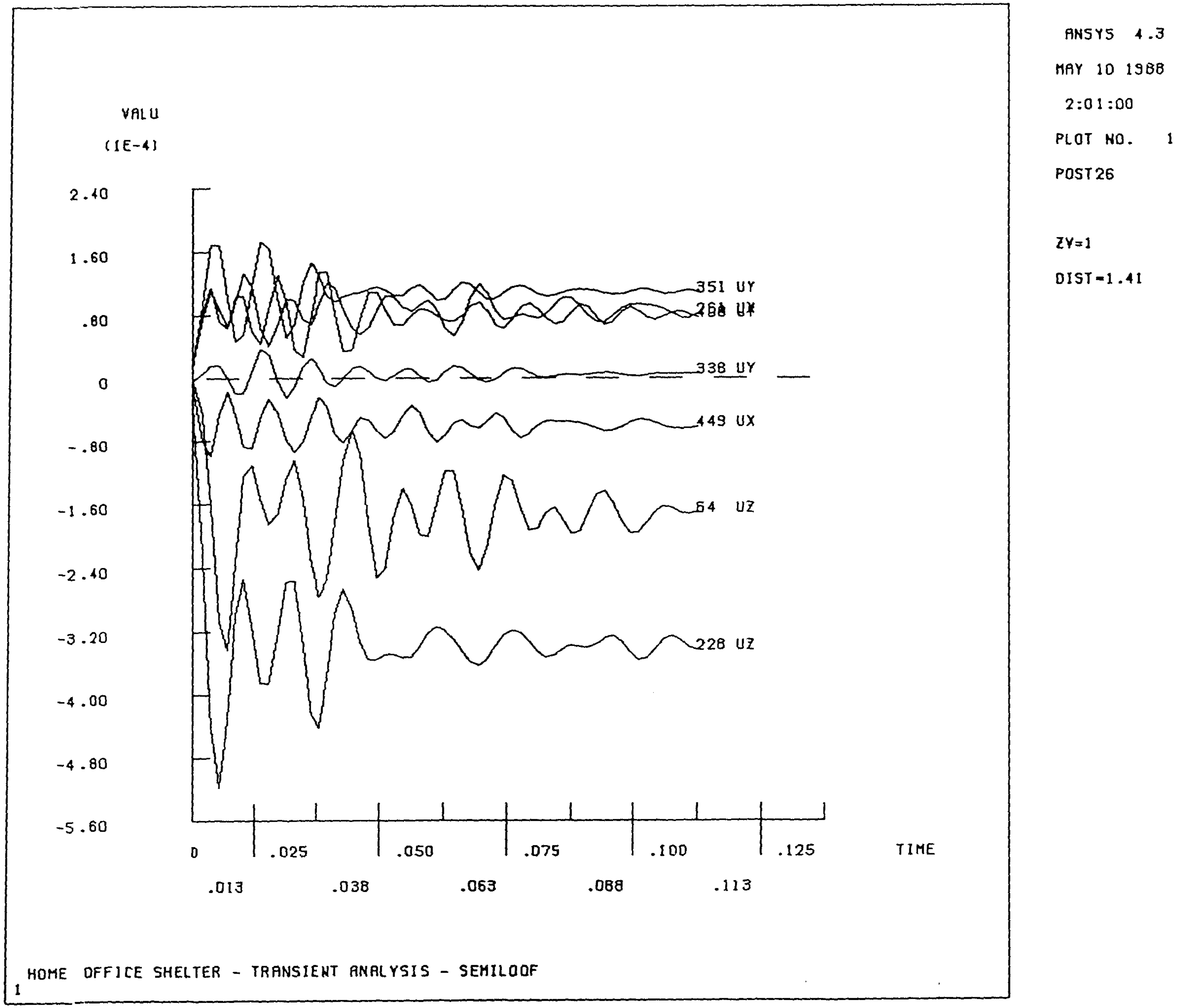


Figure 4.42 Displacements fo selected degrees of freedom vs. time.

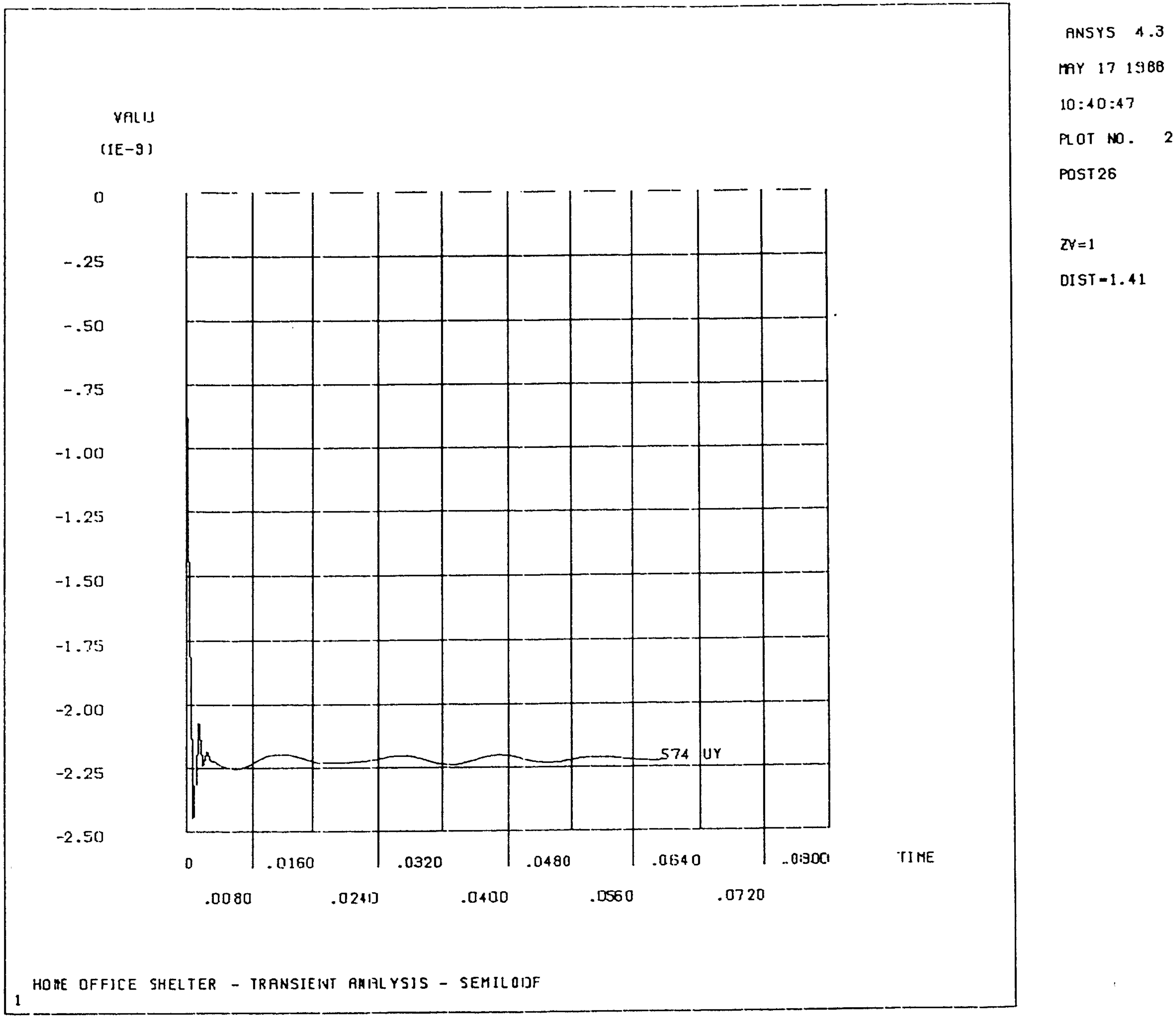


Figure 4.43 Displacement of blast door vs. time.

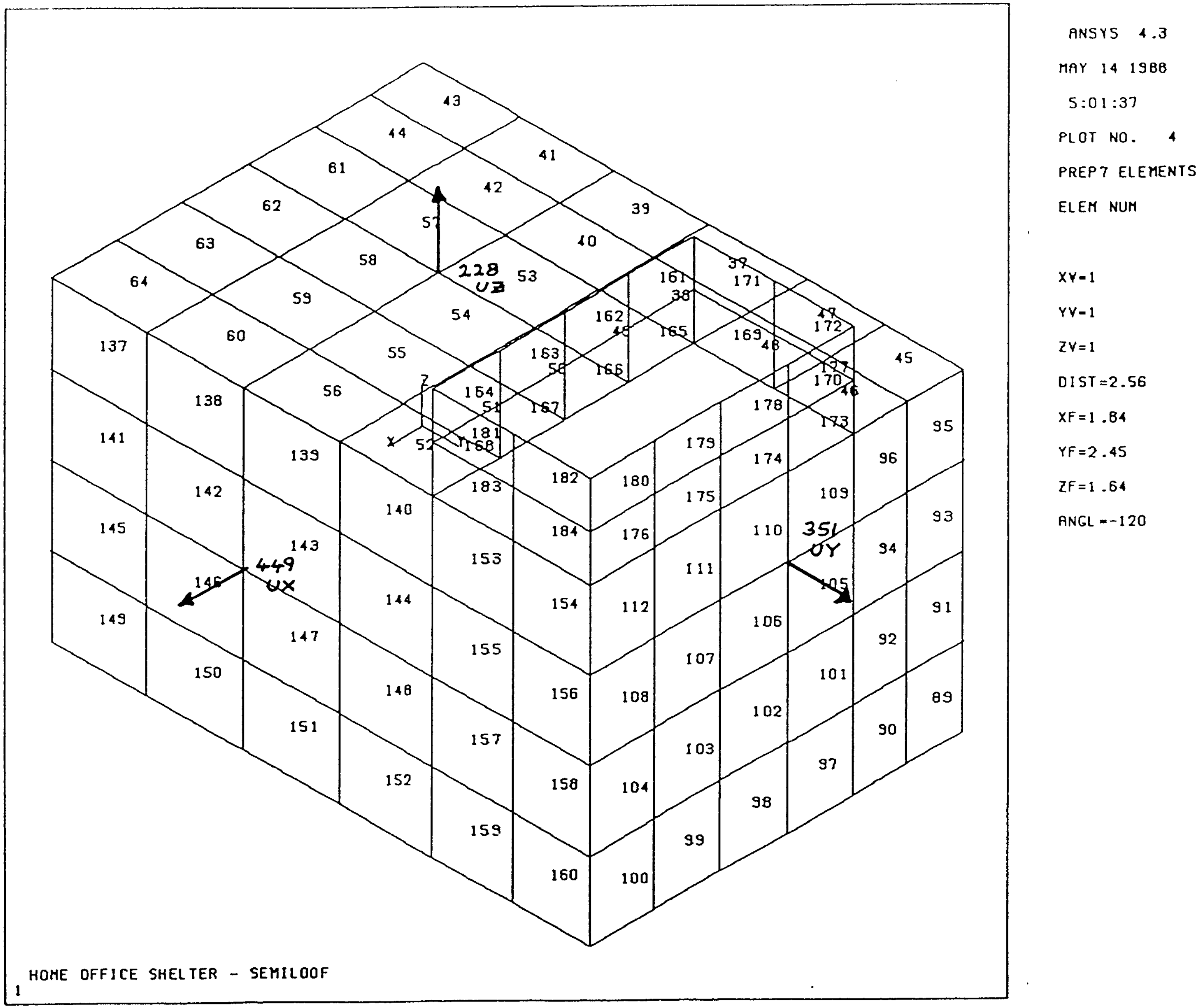


Figure 4.44 Degrees of freedom selected for transient displacement plots.

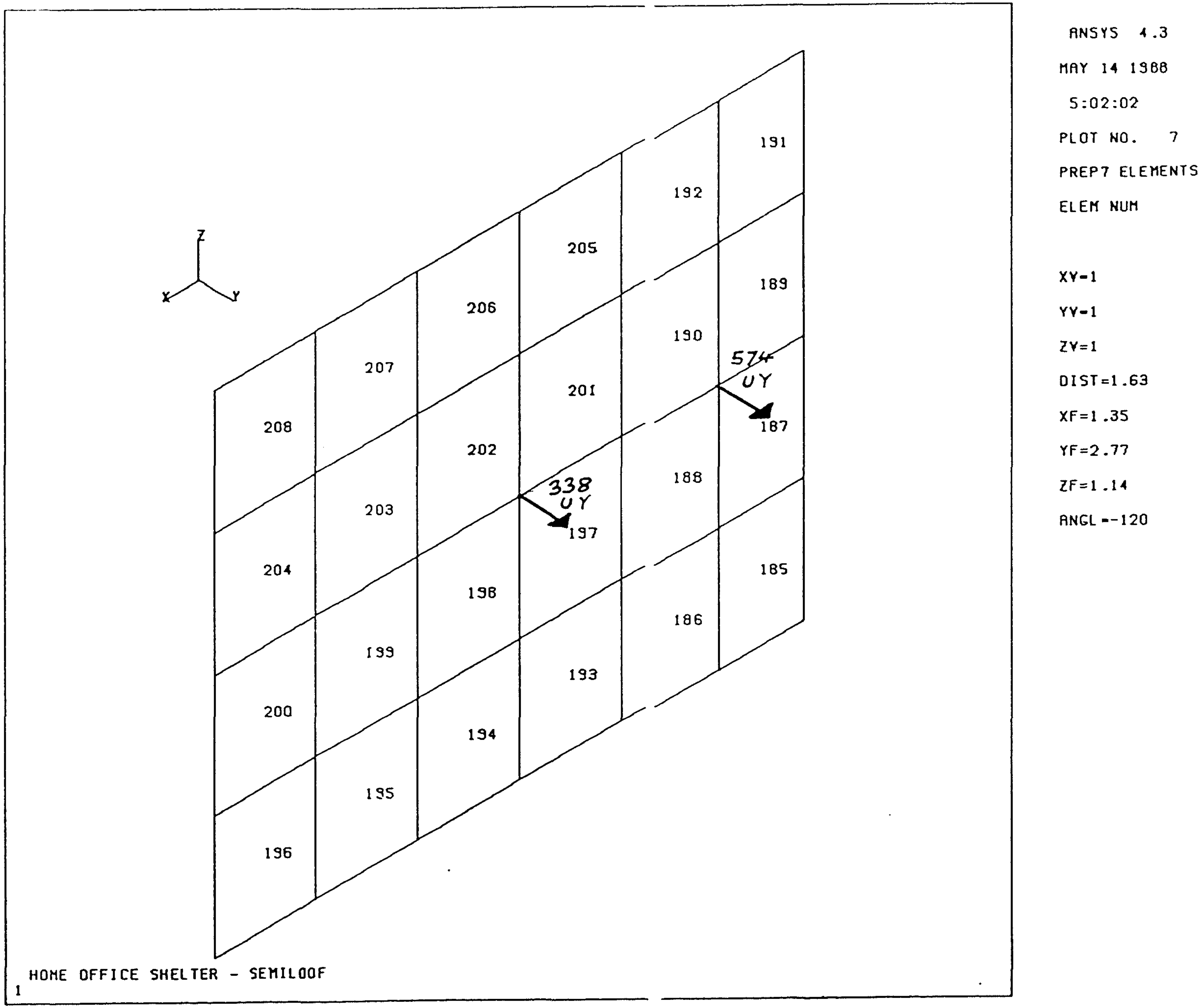


Figure 4.45 Degrees of freedom selected for transient displacement plots.

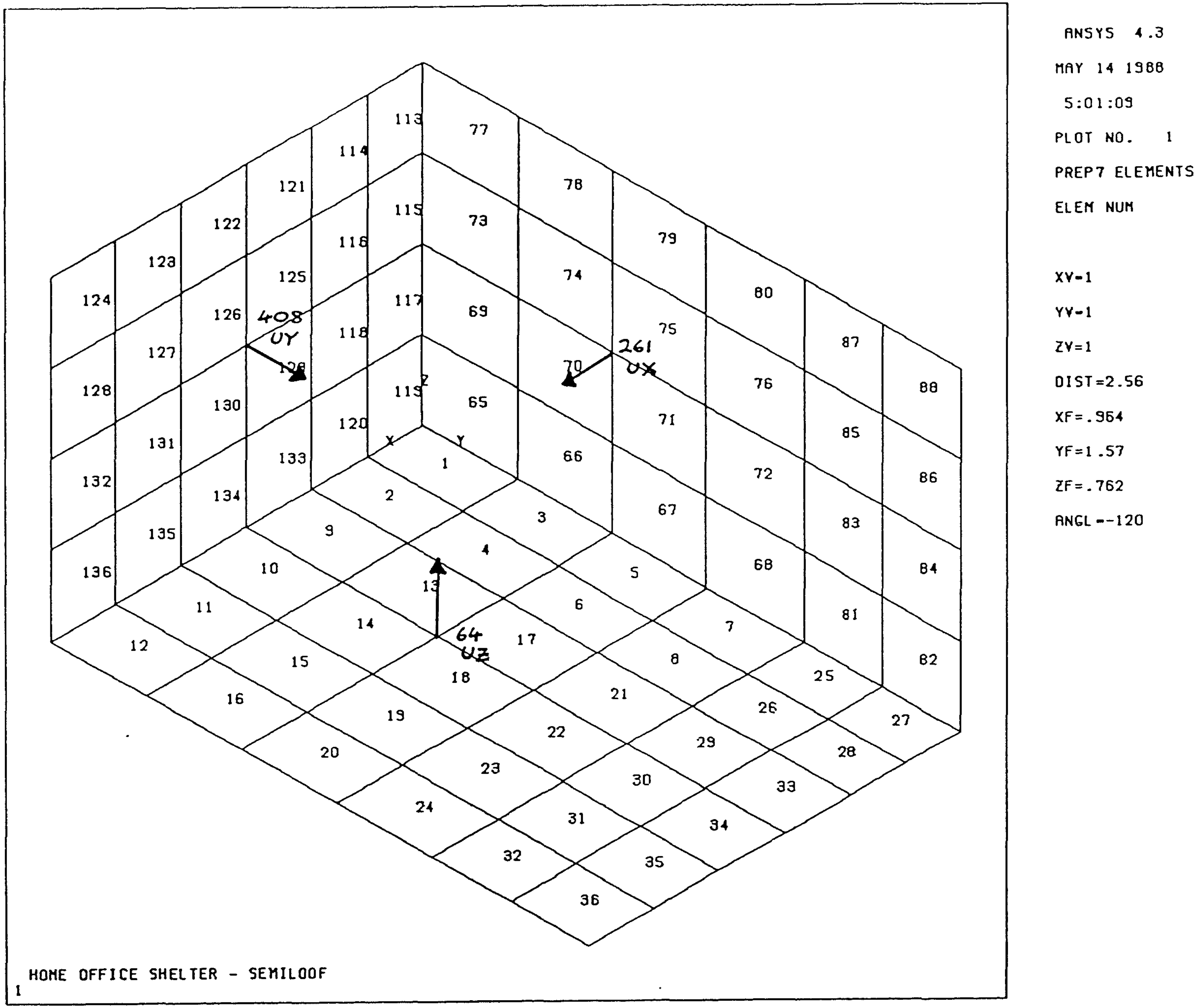


Figure 4.46 Degrees of freedom selected for transient displacement plots.

Table 4.2 Elements with bending moments greater than 35 kNm/m and 20 kNm/m from the transient analysis in the blast wall and all other panels respectively.

Element (No.)	Time (s)	Mx moment (kNm/m)	My moment (kNm/m)	Ultimate (kNm/m)
53	0.0050	24.256 (50.937)	21.115 (44.341)	51
57	0.0050	26.901 (56.492)	25.765 (54.106)	51
58	0.0050	21.736 (45.645	22.387 (47.012)	51
197	0.0055		44.798 (94.075)	90
201	0.0055		37.922 (79.636)	90
3	0.0070	24.024 (50.450)		51
4	0.0070	23.053 (48.411)		51
5	0.0070	24.956 (52.407)		51
7	0.0070	20.641 (43.346)		51
174	0.0140		25 134 (52 781)	51
175	0.0140		24 404 (51 248)	51

Figures in parentheses are factored by 2.1.

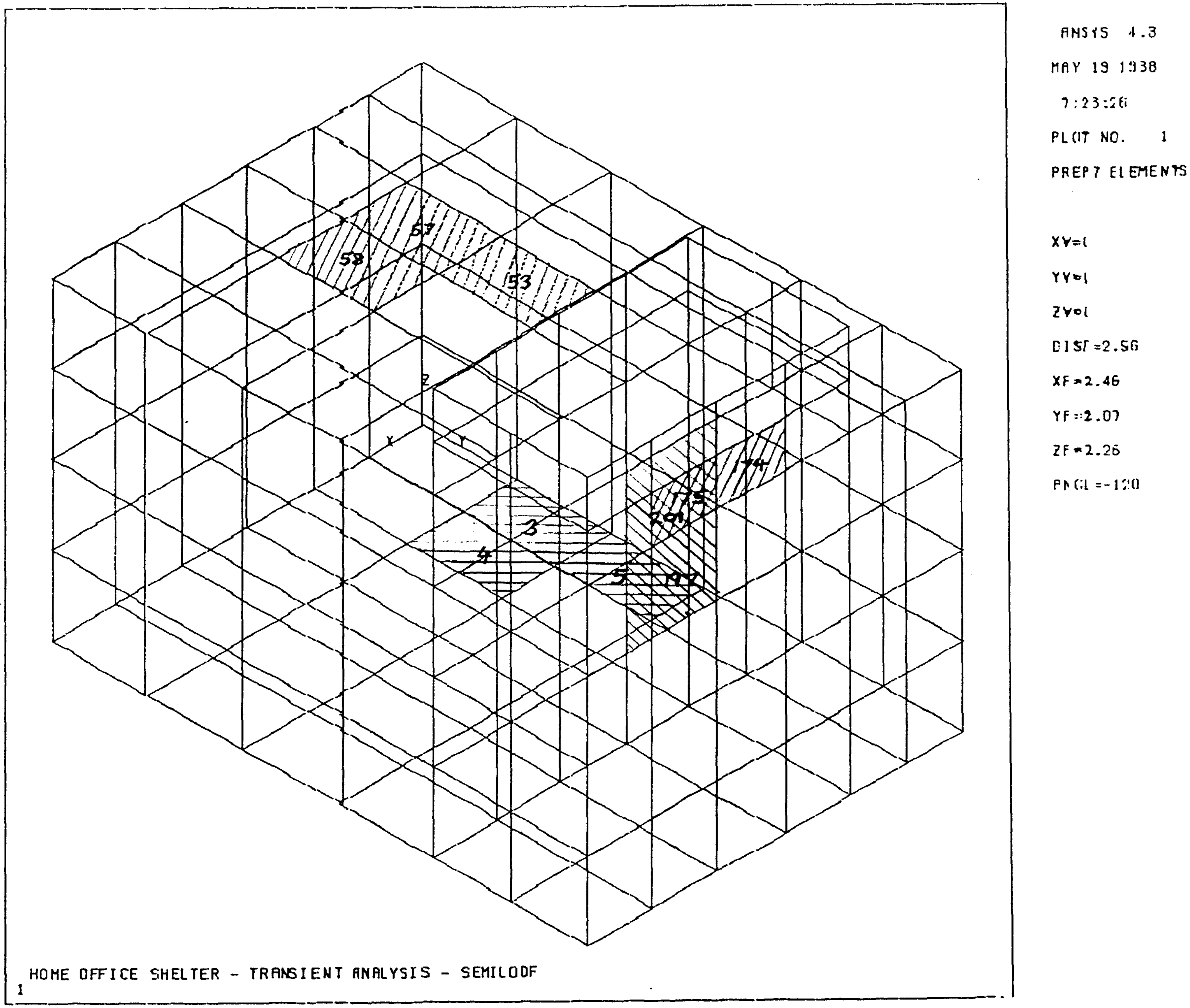


Figure 4.47 Element with bending moments greater than criteria.

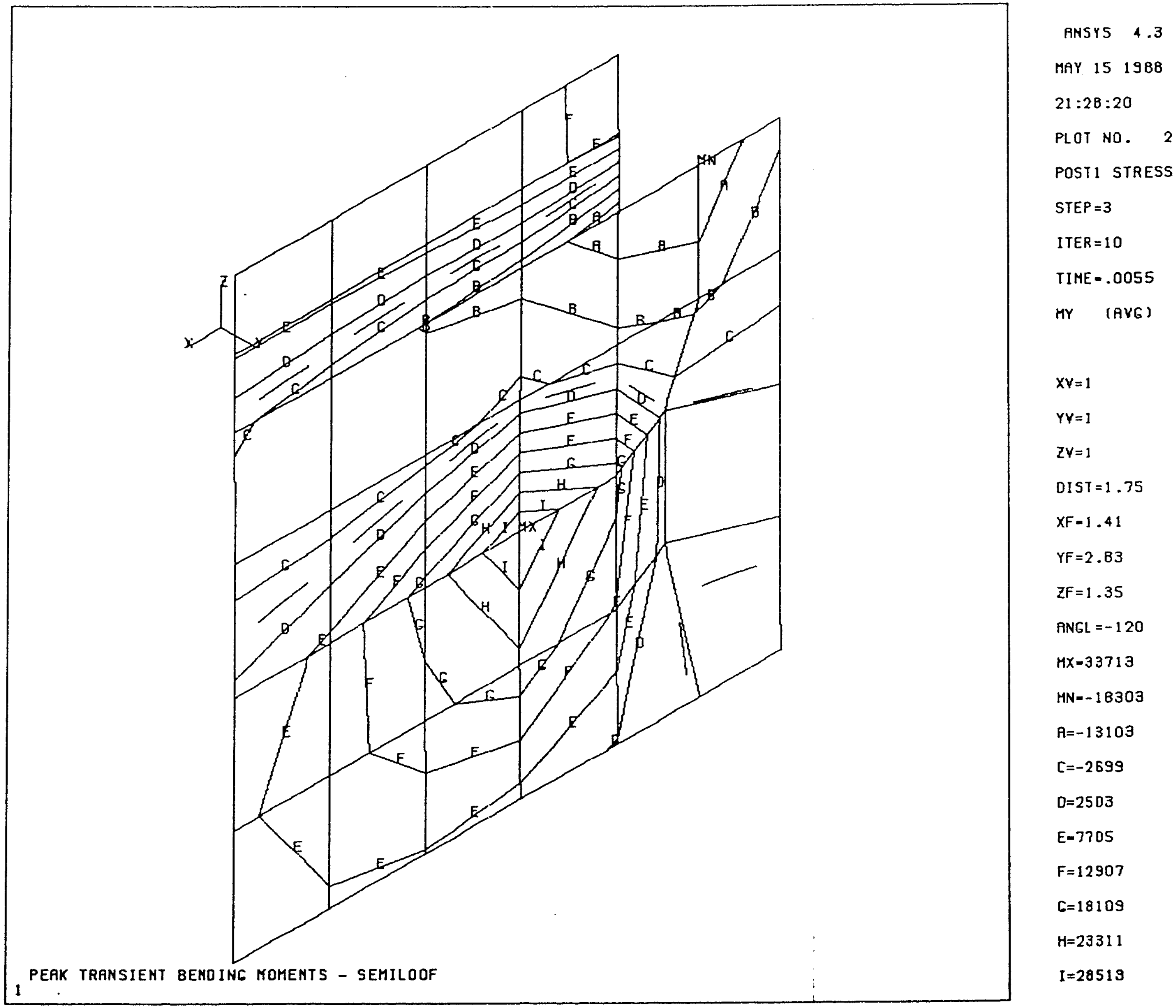


Figure 4.48 Maximum bending moments from transient analysis.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 The implemenetation of the SemiLoof shell element

The results calculated by SemiLoof in ANSYS are encouraging and agree well with both classical theory (Appendix B) and the results obtained by the ANSYS Stif93 elements, with the major advantage to SemiLoof of being able to calculate these results much faster (and in some cases more accurately) than Stif93 due to the lesser number of nodal degrees of freedom. In the modal analysis in this thesis for instance, the SemiLoof model took only 47 mins of CPU wheras the Stif93 model took 1hr 24 mins.

Unfortunately, the real potential for time saving offered by SemiLoof cannot easily be extracted by ANSYS because of the limitations of the User Element stencils (i.e. as no 5,3,5... nodal degree of freedom stencil is available the one used in this thesis was a 48x48 degree of freedom matrix (which was solved as such), although only 32 degrees of freedom were necessary for the calculations. It may be possible to improve performance by using the (ux,uy,uz,rotn) stencil with the eight node quaderilateral plot shape as shown in Fig.5.1. This would mean that rotational data from the two Loof nodes along each edge, would need to be input and ouput at the midside and corner nodes. This arrangement, although unconventional for SemiLoof, would use an optimally efficient 32x32 matrix. In its present implementation, SemiLoof forms its stiffness matrices approximately 20% quicker than Stif93; for the implementation regime mentioned above, this could feasibly be improved to 30%. Futher time savings would be gained in the displacement pass due to the solution of a 32x32 matrix rather than a 48x48. Alternatively, ANSYS's substructuring capability could be used to constrain out the unwanted degrees of freedom, though this would only be efficient for either a large model or one subject to an iterative solution.

Further improvements on the existing implementation may be the inclusion of different nodal thicknesses, orthotropic material properties and nonlinear capabilities such as plasticity and large rotation theory.

For dynamic analyses, the introduction of corresponding consistent mass and damping matrices would save the user the effort involved in substituting the SemiLoof shells for Stif93 and using its consistent mass matrix, together with Guyan Reduction, in order to calculate a lumped mass distribution facilitated by the additional use of Stif21 Lumped mass elements.

A final drawback of using SemiLoof in ANSYS is that there are no suitable beam elements available in ANSYS, to connect to the SemiLoof shells. However, the coding for such a beam element does exist in (4), and could be implemented with the User Element Utility in same manner as the SemiLoof shell has been implemented in this thesis. The SemiLoof shell can however, be used with solid and lumped mass elements.

The SemiLoof shell element is a worthwhile addition to the ANSYS element library both in its current form and also because of the further potential it offers as an efficient, alternative thin shell element.

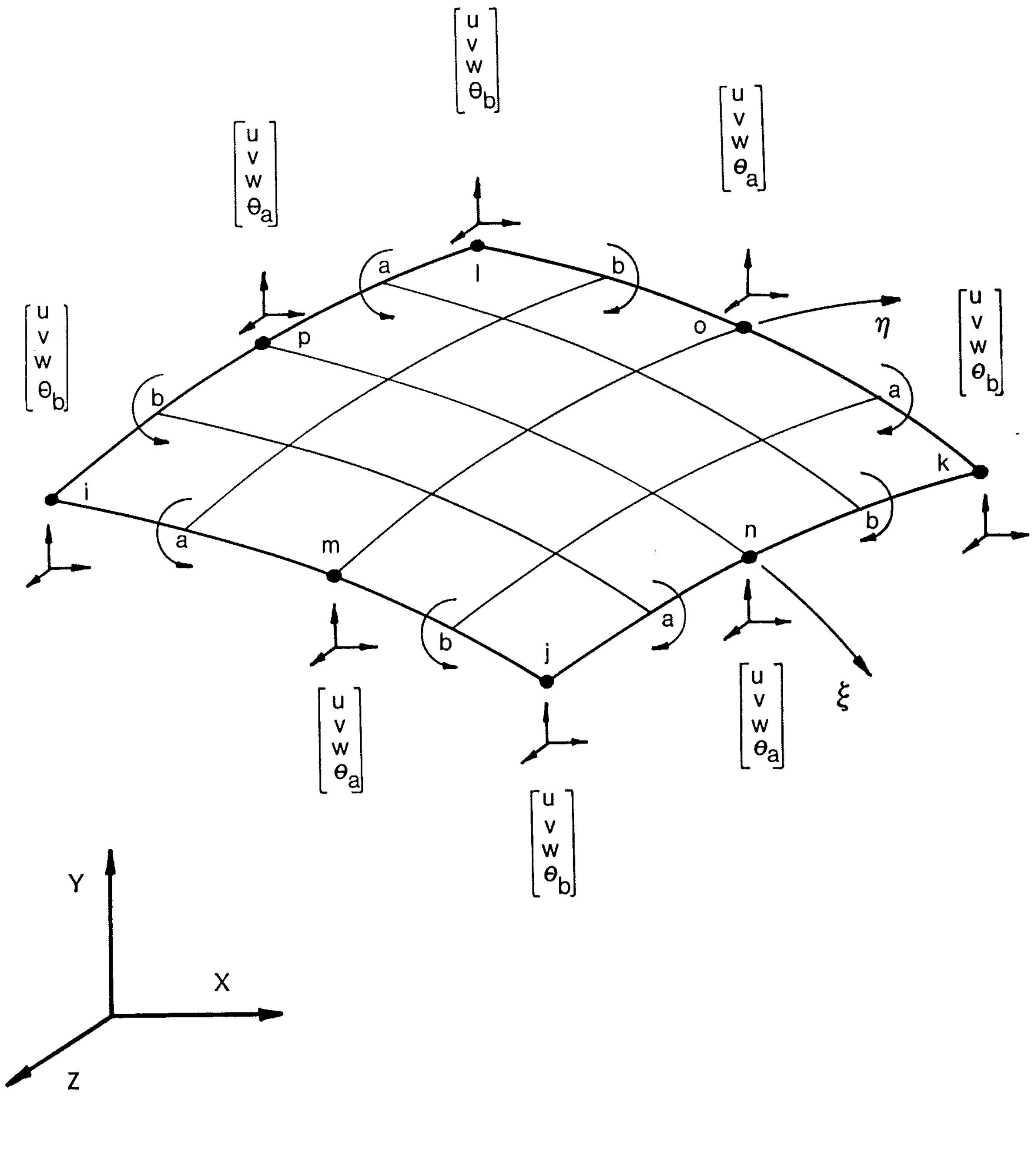


Figure 5.1 Alternative ANSYS User Element stencil utilising a 32x32 degree of freedom matrix.

5.2 Home Office Nuclear shelter - Design reappraisal

The aim of the nuclear shelter analysis was to determine whether or not the design data used by the Home Office designers was accurate with respect to the given loading data. The question of suitability of the design criteria itself was also raised, and whether or not the policy of designing the shelter as a 'one-life' protective structure was acceptable.

The analyses in this thesis served to confirm that the design data used by the Home Office analysts was adequate except for one edge of the floor as shown in Table 4.2 and Fig.4.47, where additional reinforcement is required.

An estimation of the level of damage for the calculated responses could be made by a finer re-modelling the shelter with ANSYS's reinforced concrete solid element which would take account of material nonlinearities such as cracking, crushing and plasticity of the reinforcement. Also, because it is a solid element, any internal shear deformations would be accounted for, which may be significant as the shelter cross-sectional geometry is almost on the limit of the maximum span/depth ratio recommended for thin shell theory.

The second question of the design criteria itself raises doubts about the adequacy of the design, beyond fulfilling its role as a simple blast and radiation proof structure. To the occupants of the shelter, both during and possibly after hostilities, the shelter acts as a home, and comfort and sanitation (dependant largely on the structural integrity of the shelter as discussed in Chapter 2), is of prime importance. Even after the shelter has outlived its usefulness as a blast and radiation proof structure, it may required for a much longer period as a refuge or store. Few above

ground structures are expected to remain intact after a nuclear attack and the underground shelter still be may be one of the most attractive places to take refuge in the short or medium term.
It is suggested that the shelter be re-designed to take the above points into consideration and that it should be designed as a structure which 'survives' the expected blast load intact, and remains in service for as long as necassary after that time.

As far as the general architecture of the shelter is concerned the design seems to be quite good. A second escape route from the shelter would be a welcomed addition, as reliance on the single blast door is rather an unattractive prospect. Some form of hatch in one of the walls would be adequate for this purpose. If an additional blast door were positioned at the top of the stairs, it may be possible to use the stairwell as a second room, which would provide an invaluable dimension of privacy to the occupants of the shelter. Even if the upper door failed, it would still have served to relieve some of the loading on the main door of which it is imperative in the absence of a second escape route, that it must remain in good working order. The addition of a flight of concrete stairs would serve the dual purpose of stiffening up the end wall and also providing easier access for infirmed or elderly members of the shelter party.

REFERENCES

1. DeSalvo, G.J., and Swanson, J.A.
   ANSYS User Manual.
   Swanson Analysis Systems, USA, 1985.

2. Hinton, E., and Owen, D.R.J.
   Finite Element Programming.
   Academic Press, London, 1977.

3. Irons, B. and Ahmed, S.,
   Techniques of Finite Elelements.
   Ellis Horwood Ltd, England,1980.

4. Irons, B.,
   The SemiLoof shell element.
   Finite elements for thin shells and curved members.
   Edited by Ashwell, D.G., and Gallagher, R.H.
   John Wiley and Sons, England, 1976.

5. LUSAS (London University Structural Analysis System)
   Finite Element Analysis Ltd, London.

6. BERSAFE (BERkley Structural Analysis Finite Element system)
   C.E.G.B. Berkley Nuclear Laboratories
   Gloucestershire, England.

7. Barlow J., Davies, G.A.O.,
Selected FE benchmarks for structural and thermal analyses,
NAFEMS (National Agency for Finite Element Methods and Standards)
National Engineering Laboratories
East Kilbride, Glasgow, Scotland. 1986.

8. Timoshenko S., Wionowsky-Krieger S.,
Theory of Plates and Shells.
2nd Edition, McGraw-Hill, New York, 1959.

9. Megson, T.G.H.,
Strength of Materials for Civil Engineers.
Nelson, England, 1980.

10. Stroud, D.,
Further Engineering Mathematics.
MacMillan Education Ltd.
Great Britain, 1986.

11. Domestic Nuclear Shelters - Technical Guidance,
HMSO, England, 1981.

12. Design of structures to resist nuclear weapons effects.
A.S.C.E. Manuals of Engineering Practice, No. 142.
U.S.A. 1961.

13. Roark, Raymond J., Young, Warren C.,
Formulas for Stress and Strain,
McGraw-Hill International. 1975.

14. Kohnke, P.,

ANSYS Theoretical Manual,

Swanson Analysis Systems, U.S.A. 1987.

15. Irving, D.J.,

'Large deformation elasto-plastic analysis of plates, shells and tubular joints using SemiLoof shell elements'.

Phd thesis, Kingston Polytechnic, England. 1982.

16. Crutzen, Y.,

'Non-Linear transient dynamic analysis of thin shells using the SemiLoof shell element'.

Thesis, Universite Libre De Bruxelles. 1979.

17. PAFEC Theoretical Manual (Level 5),

Strelley House, Nottingham, England. 1986.

18. Biggs, J.,

Introduction to Structural Dynamics,

McGraw-Hill. 1959.

19. Norris, C., et al,

Structural Design for Dynamic Loads,

McGraw-Hill. 1964.

20. Ormerod R.,

Nuclear Shelters - A guide to design

Architectural Press Ltd. 1983

APPENDIX A

THE ANSYS USER ELEMENT UTILITY - SEMILOOF AND ANSYS INTERFACING SUBROUTINES

A.1 Preprocessing

The ANSYS general database preprocessor PREP7, is a program whose primary objective is to organise analysis input data and write a formatted analysis data file, FILE27.

The finite element model generated in PREP7 includes model geometry, material properties, applied loading data and displacement constraints. Additional items of information included at this stage are the element types, analysis type and the actuation of other capabilities such as plasticity or large rotation theory.

If ANSYS is executed interactively, information submitted to PREP7 may be listed or plotted in real time for inspection.

The organisation of a simple ANSYS analysis, showing primary routines and files is shown in Table 1.5.

Once the model data has been inspected and deemed satisfactory, all data currently in the preprocessing database is re-organised and written to the analysis input file FILE27, which contains all the analysis data needed by the solution routine of ANSYS.

The main advantage of using PREP7 to create FILE27, rather than typing it in directly, is that the model database can be built up interactively. Hence information can be listed, plotted and generally viewed as it is input, resulting in the immediate identification and correction of errors.

To the user element programmer, the consequences of using PREP7 to create the analysis file are as follows:

a. In order to create plots of the user element, PREP7 must have access to information regarding the element shape (e.g. beam, shell, solid..., linear, triangular, quad etc.).

b. User element data input to PREP7 must be checkable.

Each element type has a unique spreadsheet defined for it within PREP7 which expects different information as the element type dictates. For example, PREP7 needs to know how many nodes the element has, or the degree of freedom set at each node, or the number of constants required to serve the supporting element theory.

Hence the user element programmer must define the spreadsheet for the new, user implemented, element type.

By using this spreadsheet approach, ANSYS can check for superflous, incorrect or missing data, which can then be modified before FILE27 is written.

The spreadsheet and plot information referred to must be made accesible to PREP7 as soon as the element type is specified in order that a spreadsheet for that element type can be called and used to check the input data.

The spreadsheet information required by the preprocessor for the user element is contained in the two subroutines, USEREL and USERPT, which are linked to ANSYS before execution commences.

Therefore, if a new element type is to be defined, the supporting subroutines USEREL and USERPT must be revised.

USEREL and USERPT define the following parameters:

USEREL - DEFINES THE PARAMETERS FOR THE USER ELEMENT

------ (Called by PREP7)

Number of physical element dimensions.

Nodal degree of freedom set.

Symmetric or unsymmetric matrices.

Nodal degrees of freedom for nodal transformations.

Number of nodes.

Number of pressures.

Number of user defined variables to be associated with each unique element.

Dimensions of element matrices.

Linear or non-linear element.

Structural or thermal element.

USERPT - DEFINES THE PARAMETERS FOR THE USER ELEMENT PLOTS

------ (Called by PREP7 and POST1)

Element shape.

Number of nodes.

When preprocessing is complete, and the analysis file write command is executed, ANSYS checks, with USEREL, that all user element information has been provided.

FILE27 is then written which contains the analysis data.

A.2 Finite Element Solution

The solution phase can be defined thus:

1. Generation of elemental stiffness, mass and damping matrices together with load vectors and stress matrices.
2. Solution of stiffness equations for displacements.
3. Calculation of stresses and other post displacement items.

In this thesis the work done in implementing the SemiLoof shell element has lain almost completely within the domain of steps 1 and 3, relying on the host program ANSYS to take care of pre and postprocessing and the equation solution.

The following discussion aims to indicate the ANSYS program modules which are affected by the implementation of a foreign stiffness and stress calculation utility.

ANSYS reads FILE27 when it wishes to excecute an analysis.

FILE27 has been produced by the preprocessor PREP7 and contains all the information required for that particular analysis.

This information includes:

1. Element type list. e.g. beams, shells etc.
2. Nodal numbers and corresponding geometric locations.
3. Element nodal definitions.
4. Material properties.
5. Boundary contraints.
6. Applied loads.
7. Analysis options. ie additional supporting information required for a dynamic or nonlinear etc. analysis.

All items except Nos. 6 & 7 in the above list are required for the generation of the element stiffness matrices.

ANSYS has an element library containing many different element types, each of which requires a unique set of supporting information.

Once the element type has been identified, subroutines supporting that element type are called upon to process the incoming data and generate the element stiffness matrix.

Items 2 to 5 in the above list are read by the element subroutines which then generate the sub matrices ([B], [D]) and hence the stiffness matrix [K].

Since the sub-matrices [D] and [B] are both available at this stage, they are multiplied together to form the stress matrix which is used in the stress pass for stress calculations.

Additionally, because the area of the element has been calculated to service stiffness matrix generation, the pressure loads are resolved into nodal loads.

Any other required element matrices such as mass or damping, are also generated here.

The stiffness matrices and pressure vectors are written to FILE02 whilst the stress matrices are written to FILE03 via the SVR array if required.

The calculated stiffness matrices are then submitted for displacement solution, followed by nodal force and element stress calculations. All the calculated information is then passed to FILE12 for reading by the postprocessor POST1.

A.2.1 Stiffness matrix generation

The isoparametric stiffness matrix is calculated:

$$[K] = \int_{-1}^{1}\int_{-1}^{1} [B]^t [D] [B] \det J \, d\xi \, d\eta$$

where the matrices to be calculated are:

[B] Shape function derivitives

[D] Elasticity matrix

det J Jacobian determinant

[K] Element stiffness matrix

The following discussion will outline the programming methodology which ANSYS uses in the User Element Utility.

Schematically, the task of programming to calculate the element stiffness matrices can efficently be carried out using the programming loops shown in Table A.1.

ANSYS begins the solution by first reading FILE27 and putting all the analysis data into COMMON storage, which makes this information available to the appropriate subroutines.

ANSYS then calls subroutine ST100 whose job is to return a set of element matrices to the main program.

It is ST100 which contains the coding for the production of the element matrices, implemented by the User Element Programmer.

Element geometry and material properties are passed into ST100 by three routes:

1. The arguments in the ST100 subroutine call.
2. The arguments in a call to subroutine GETELD which reads and writes to FILE03.
3. Through COMMON storage.

Table A.1 Methodology employed to create the elemental stiffness matrix.

```
 --> 1. ELEMENT LOOP
 :       Read element geometry and material properties.
 :       Zero the current element stiffness matrix.
 :       Calculate the elasticity matrix [D].
 :
 :
 : -> 2. INTEGRATION POINT LOOP
 ::
 ::        Read local coordinates of integration points.
 ::        Formulate shape functions and local shape function
 ::        derivitives.
 ::
 ::        Formulate the Jacobian determinant and inverse.
 ::
 ::
 ::        Formulate strain matrix terms.
 ::        Assemble strain matrix [B].
 ::        Formulate [B][D].
 ::        Formulate [B][D][B]det J and integrate.
 ::        Assemble into the element stiffness matrix [K].
 ::-----< Assemble [B][D].
 :
 :
 :         Write element stiffness matrix [K] to a file
 :------< Write [D][B] to a file
```

A.2.1.1 ST100 - CALCULATES ELEMENT MATRICES

----- (Called by the main program ANSYS).

The information passed to and from the main program includes:

Element number.

Element type (STIF100).

Matrices calculated.

Element matrices.

The calculated element matrices which are passed back to the main program via the ST100 calling arguments are then written to FILE02 for use in the solution phase.

As well as calculating the element matrices, the sub-matrices [D] and [B] are multiplied together and passed in the SVR array, via PUTELD, to FILE03 for use in the ANSYS stress pass.

A.2.2 Solution of equations

Once the elemental matrices have been formed, the basic equations can be solved.

e.g. $[M].\{\ddot{u}\} + [C].\{\dot{u}\} + [K].\{u\} = \{F(t)\}$

or in the case of statics:

$$[K].\{u\} = \{F\}$$

where [K] is the global stiffness matrix

{u} is a column vector of nodal displacements

{F} is a column vector of nodal forces

There are three steps to the solution procedure:

1. Assemble the global stiffness matrix.
2. Assemble the global nodal force vector from its constituent parts, e.g. nodal forces, pressure loads.
3. Solve the equations for displacaments.

After all the matrices for all the elements have been formulated and written to FILE02 the displacement solution commences.
The solution method employed by ANSYS is a frontal solver which solves the equations on a per element basis and writes the global triangularised matrix to FILE11.
After the equations have been fully reduced, FILE11 is then read and backsubstitution commences. The displacement solution is then written to FILE12.

A.2.3 Stress solution

The stress pass commences by reading the displacement file on a per element basis, and multiplying with [D][B].
Nodal forces are also calculated at this stage multiplying [K]{u} on a per element basis to solve for {F}.
SR100 is the user subroutine in which stresses are calculated.
SR100 reads [D][B] from FILE03 via GETELD (e.g. [D][B] at each integration point) and the displacements from FILE12 to calculate the element stresses.
The calculated element stresses are labeled and passed to FILE12 to be read by the postprocessor POST1.

A.2.3.1 SR100 - CALCULATES ELEMENT STRESSES
----- (Called by the main program ANSYS).

The information passed to and from the main program includes:
Element number.
Element type (STIF100).
Nodal displacements.

The [D][B] matrix for each integration point is passed into SR100 from FILE03 and stacked into a full elemental [D][B] matrix which is then multiplied by the element nodal displacements in order to calculate the element forces and moments at each integration point. From the basic forces and moments, other items can be calculated such as average centriodal stresses and principal stresses.

SEMILOOF AND ANSYS INTERFACING SUBROUTINES

```
C    ***********************************************************
C    *  USERS IS A FILE CONTAINING THE ANSYS USER ELEMENT        *
C    *  SUBROUTINES, NOT INCLUDING THE STRESS PASS               *
C    *  ------------------------------------------------------- *
C    *  SUBROUTINES INCLUDED ARE :    USEREL                     *
C    *                                USERPT                     *
C    *                                ST100                      *
C    *                                USERFN   (EMPTY)           *
C    *                                USERIT   (EMPTY)           *
C    *                                USERLD   (EMPTY)           *
C    ***********************************************************
C
C ------------------------------------------------------------
C
C
C    ***********************************************************
C    *  SUBROUTINE  USEREL                                       *
C    ***********************************************************
C
C
      SUBROUTINE USEREL (ITYP,IPARM,KYSUB,KEY3D,KDOF,KUNSYM,KTRANS)
C ******** DEFINE PARAMETERS FOR ANSYS USER ELEMENT ***********
C ***** FORTRAN SYNTAX ON DOUBLE PRECISION STATEMENT                  SYSTEM
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                             SYSTEM
C
      INTEGER IPARM(20,12),KYSUB(9),ITYP,JTYPE,KEY3D,KDOF,KUNSYM,KTRANS
C
C    ****** DETERMINE TYPE OF ELEMENT AND THEN BYPASS IF NOT USER ELEMENT *****
      JTYPE = IPARM(ITYP,3)
      IF (JTYPE .NE. 100) GO TO 100
C     NOTE: THE BELOW USER ELEMENT DEFINITIONS MAY BE FUNCTIONS OF
C           KYSUB(INPUT QUANTITY KEYOPT)
C
C          **********  SET 3-D KEY  **********
C     IF ELEMENT GEOMETRY IS DEFINED IN 2-D, KEY3D = 0
C     IF ELEMENT GEOMETRY IS DEFINED IN 3-D, KEY3D = 1
      KEY3D = 1
C
C          **********  DEFINE DOF SET AT EACH NODE  **********
C          KDOF=0  UX,UY,UZ,ROTX,ROTY,ROTZ
C          1-UX  2-UY  3-UZ  4-ROTX  5-ROTY  6-ROTZ
C          7-PRES  8-TEMP  9-VOLT  10-MAG
C          11-UX,UY  12-UX,UY,ROTZ  13-UZ,ROTX,ROTY
C          14-UX,UY,UZ  15-PRES,TEMP  16-ROTX,ROTY,ROTZ
C          17-UX,UZ  18-UX,UY,UZ,ROTZ  19-TEMP,VOLT,MAG
C          20-UX,UY,PRES  21-UX,UY,UZ,TEMP,VOLT,MAG
      KDOF = 0
C
C          **********  SET UNSYMMETRIC MATRIX KEY  **********
C          KUNSYM = 0 PRESUMES SYMMETRIC MATRICES
C          KUNSYM = 1 PRESUMES UNSYMMETRIC MATRICES
C             IF MATRICES ARE UNSYMMETRIC FOR USER ELEMENT, KEYOPT(8)
C              MUST ALSO BE SET TO 1, WHICH HAS BEEN RESERVED FOR
C              THIS PURPOSE.
      KUNSYM = 0
C
C      *****  DEFINE PATTERN FOR ELEMENT TO GLOBAL TRANSFORMATION  ***
```

```
C      NOTE:  ALL ELEMENTS MUST BE GENERATED IN THE GLOBAL CARTESIAN
C             SYSTEM.  HOWEVER, THE USER MAY BE USING A NODAL SYSTEM
C             WHICH IS DIFFERENT FROM THE GLOBAL CARTESIAN SYSTEM
C             (E.G. NROTATE COMMAND).  KTRANS PERMITS THE PROGRAM
C             TO PROPERLY ROTATE THE DEGREES OF FREEDOM.
C          0-NO NODE ROTATION              1-UX,UY
C          2-UX,UY,UZ,ROTX,ROTY,ROTZ       3-UX,UY,UZ
C          4-UZ,ROTX,ROTY                  5-UX,UY,UZ,ROTZ
C                                          7-UX,UY,UZ,-,-,-(3 DOF NOT
C                                                          TRANSFORMED)
      KTRANS = 7
C
C          **********  DEFINE NUMBER OF NODES  **********
      IPARM(ITYP,8) = 8
C
C          ***** DEFINE NUMBER OF TEMPERATURES (DELTEM,TEMPER) ******
C                   (FLUENCES MAY BE INCLUDED WITH THE TEMPERATURES)
C         USE MAXIMUM OF EITHER ELEMENT TEMPERATURES OR NODAL TEMPERATURES
C       FOR HEAT TRANSFER ANALYSES, NUMBER OF HEAT GENERATION RATES
      IPARM(ITYP,11) = 1
C
C          ***** DEFINE NUMBER OF PRESSURES (PRESS) *******
C               IF THERMAL ANALYSIS, TWO TIMES NUMBER OF CONVECTION SURFACES
      IPARM(ITYP,6) = 1
C
C          ***** SET ZEROED VARIABLES (NOITUEP)
      IPARM(ITYP,12) = 0
C
C          ***** DEFINE NUMBER OF REAL CONSTANTS FOR ELEMENT (RVR) ********
      IPARM(ITYP,10) = 4
C
C          ***** DEFINE NUMBER OF VARIABLES TO BE SAVED (SVR) *****
      IPARM(ITYP,7) = 816
C
C          ***** DEFINE NUMBER OF ROWS IN ELEMENT MATRICES (KTIK) *****
C                 THIS VALUE IS DETERMINED BY MULTIPLYING THE NUMBER OF
C                 NODES BY THE NUMBER OF DEGREES OF FREEDOM PER NODE
C                 (=NUMROW(ITYP)).  I.E.  FOR STIF60, KTIK = 18,
C                 BECAUSE IT HAS THREE NODES.  MAXIMUM VALUE = 60
      IPARM(ITYP,9) = 48
C
C          **********  SET KEY TO IDENTIFY NON-LINEAR ELEMENT  **********
C             NONLINEAR ELEMENT IMPLIES THAT THE MATRICES WILL BE REFORMED
C             EVERY ITERATION, REGARDLESS OF OTHER INFORMATION.
C          0-LINEAR ELEMENT     1-NONLINEAR ELEMENT
      IPARM(ITYP,4) = 0
C
C          **********  SET KEY FOR THERMAL ELEMENT (KAN,-1)  **********
C        IPARM(ITYP,1) = 0  ELEMENT MAY ONLY BE USED IN A STRESS ANALYSIS
C        IPARM(ITYP,1) = 1  ELEMENT MAY ONLY BE USED IN A THERMAL ANALYSIS
C               THERMAL ANALYSIS IS DEFINED AS KAN,-1 OR THERMAL
C               SUBSTRUCTURE ANALYSES.  STRESS ANALYSES ARE DEFINED
C               AS ALL OTHER ANALYSES.
      IPARM(ITYP,1) = 0
  100 RETURN
      END
C
C
```

```
C
C    ***************************************************************
C    *  SUBROUTINE  USERPT                                        *
C    ***************************************************************
C
C
      SUBROUTINE USERPT (INODE,JTYPE,KSHAPE,NNODE)
C **********  USER SUBROUTINE FOR ANSYS PLOT SHAPE  **********
C ***** FORTRAN SYNTAX ON DOUBLE PRECISION STATEMENT                 SYSTEM
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                            SYSTEM
C *** ANSYS(R) COPYRIGHT(C) 1971,78,82,83,85,87  SWANSON ANALYSIS SYST. INC. ***
C         DEFINE ELEMENT SHAPE AND NUMBER OF NODES, FOR PLOTTING
      INTEGER INODE(20),JTYPE,KSHAPE,NNODE
C     *********BYPASS IF NOT USER ELEMENT (JTYPE = 100) *********
      IF (JTYPE .NE. 100) GO TO 100
C     ****** SELECT SHAPE TO BE PLOTTED BY SETTING KSHAPE *******
C      KSHAPE =  0 - NO PLOT
C      KSHAPE =  2 -  2 NODE LINE
C      KSHAPE =  3 -  3 NODE TRIANGLE
C      KSHAPE =  4 -  4 NODE QUADRILATERAL
C      KSHAPE =  5 -  8 NODE 3-D SOLID
C      KSHAPE =  6 -  8 NODE QUADRILATERAL
C      KSHAPE =  7 - 20 NODE 3-D SOLID
C      KSHAPE = 10 - 16 NODE 3-D SOLID
C      KSHAPE = 11 -  4 NODE TEE
C      KSHAPE = 12 - 10 NODE TETRAHEDRON
C      KSHAPE = 13 -  6 NODE TRIANGLE
      KSHAPE = 6
C          **********  SET NUMBER OF ACTUAL NODES  **********
      NNODE = 6
 100  RETURN
      END
C
C
C
C    ***************************************************************
C    *  SUBROUTINE  ST100                                         *
C    ***************************************************************
C
C
      SUBROUTINE ST100 (IELNUM,ITYP,KELIN,KELOUT,NR,KTIK,ZS,ZASS,DAMP,
     1    GSTIF,ZSC)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      EXTERNAL TRACK,GETELD,PUTELD,PROPEV,NONTBL,VZERO,MHTCH         EXTERNAL
      INTEGER LP(3)
C
C    ***** STCOM STORAGE *****
C
      INTEGER   IELNUM,ITYP,KELIN(6),KELOUT(6),NR,KTIK,
     1 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,KNLRST,
     2 K13,NPRPVL,MATST,K5,K16,IPROP,KCPDS,
     3 K20,KAY,MODE,ISYM,KAHD,IDEBUG,IXXX,
     4 ITYPE,MAT,IELEM,NROW,JTYPE,IPLOT,IPRINT,KTEMTP,KCONCV,KBICNV,
     5 KEYPLS,KEYCRP,KEYSWL,KYSUB(9),K21,NODES(20),    EPAR(50)
      REAL   ERRVAR(5)
      DOUBLE PRECISION                                               DPDCLR
     1 DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     2 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
```

```
     3 ACEL,OMEGA,CGOMEG,CGLOC,DXXX,
     4 ELMASS,XCENTR,YCENTR,ZCENTR,TFCP,SUBEX,  ERPAR(20),
     5 XYZEQ(20,3),X(20),Y(20),Z(20),  ELVOL
      COMMON /STCOM/ DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     1 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     2 ACEL(3),OMEGA(6),CGOMEG(6),CGLOC(3),        DXXX(16),
     3 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,KNLRST,
     4 K13,NPRPVL,MATST,K5,K16,IPROP(20),KCPDS,
     5 K20,KAY(10),MODE,ISYM,KAHD,IDEBUG(10),       IXXX(41)
      EQUIVALENCE  (ITYPE,EPAR(1)), (MAT,EPAR(2)), (IELEM,EPAR(5)),
     1 (NROW,EPAR(7)), (JTYPE,EPAR(11)), (IPLOT,EPAR(12)),
     2 (IPRINT,EPAR(13)), (KTEMTP,EPAR(14)), (KCONCV,EPAR(16)),
     4 (KBICNV,EPAR(17)), (KEYPLS,EPAR(18)), (KEYCRP,EPAR(19)),
     5 (KEYSWL,EPAR(20)), (KYSUB(1),EPAR(21)), (K21,EPAR(30)),
     6 (NODES(1),EPAR(31))
      EQUIVALENCE  (ELMASS,ERPAR(1)), (XCENTR,ERPAR(2)),
     1 (YCENTR,ERPAR(3)), (ZCENTR,ERPAR(4)), (TFCP,ERPAR(5)),
     2 (SUBEX,ERPAR(6))
      EQUIVALENCE (X(1),XYZEQ(1,1)),(Y(1),XYZEQ(1,2)),(Z(1),XYZEQ(1,3))
C
C    ***** END OF STCOM STORAGE *****
C
      DOUBLE PRECISION                                                  *UPD*
     1 ZS(KTIK,1),ZSC(KTIK),
     3 PROP(3)
      DOUBLE PRECISION                                                  *UPD*
     1 DELTEM(2),TEMPER(2),PRESS(3),
     2 RVR(4),SVR(816),
     4 U(240), AVETEM
      DATA LP / 1, 3, 10/
      CALL TRACK(5,'ST100 ')                                            TRACK
      CALL GETELD (IELNUM,ITYP,EPAR(1),ERPAR(1),DELTEM(1),TEMPER(1),    GETELD
     1 PRESS(1),CON,RVR(1),SVR(1),XYZEQ(1,1),U(1))
      CALL PROPEV (IELEM,MAT,JTYPE,LP(1),AVETEM,PROP(1),3)              PROPEV
C
C        ******* STIFFNESS MATRIX ********
C
      CALL INTERIN (NODES, X, Y, Z, PROP, PRESS, RVR,
     ~ SVR, ZS, ZSC)
      KELOUT(1) =  1
C
C           ******* LOAD VECTOR *******
C
      KELOUT(5) = 1
C
      CALL PUTELD (IELNUM,EPAR(1),ERPAR(1),CON,SVR(1))                  PUTELD
      CALL TRACK( 15,'ST100 ')                                          TRACK
      RETURN
      END
C
C
C
C    ***************************************************************
C    *  OTHER ANSYS SUBROUTINES                                     *
C    ***************************************************************
C
C
      SUBROUTINE USERFN
```

```
C     **** THIS SUBROUTINE IS ALWAYS CALLED AFTER ALL LOAD STEPS ****
C     INPUT IS THRU FILES
C     OUTPUT IS THRU FILES (INCLUDING PRINT FILE)
      RETURN
      END
C
      SUBROUTINE USERIT
C     **** THIS SUBROUTINE IS ALWAYS CALLED AFTER EACH ITERATION ****
C     INPUT IS THRU FILES
C     OUTPUT IS THRU FILES (INCLUDING PRINT FILE)
      RETURN
      END
C
      SUBROUTINE USERLD
C     **** THIS SUBROUTINE IS ALWAYS CALLED AFTER EACH LOAD STEP ****
C     INPUT IS THRU FILES
C     OUTPUT IS THRU FILES (INCLUDING PRINT FILE)
      RETURN
      END
C
C************************************************************************
```

```
C    ***************************************************************
C    * INTERFACE IS A FILE CONTAINING THE SUBROUTINES WHICH SERVE *
C    * TO INTERFACE ANSYS WITH THE SEMILOOF SUBROUTINES           *
C    * ---------------------------------------------------------- *
C    * SUBROUTINES INCLUDED ARE :  INTERIN                        *
C    *                             INTEROUT                       *
C    ***************************************************************
C
C
C --------------------------------------------------------------
C
C
C    ***************************************************************
C    *  SUBROUTINE  INTERIN                                       *
C    ***************************************************************
C
C
      SUBROUTINE INTERIN (NODES,X,Y,Z,PROP,PRESS,RVR,
     ~   SVR,ZS,ZSC)
C     ===============
C
C     THIS SUBROUTINE PROVIDES THE INPUT INTERFACE BETWEEN ST100 AND
C       SEMILOOF SUBROUTINES
C
      DOUBLE PRECISION X(20),Y(20),Z(20),PROP(3),PRESS(1),
     ~   RVR(4), SVR(816), ZS(48,48), ZSC(48,1)
C
      DOUBLE PRECISION COORD(2000,3), VPROP(5,1), ELSTIF(528),
     ~ ELOAD(32,1), ELOAD1(48,1)
C
      INTEGER LNODS(8,1), NODES(20)
C
C       ******* SWAP ANSYS ELEM NODE ORDER FOR SEMILOOF DEFINITION ******
C
      LNODS(1,1)=NODES(1)
      LNODS(2,1)=NODES(5)
      LNODS(3,1)=NODES(2)
      LNODS(4,1)=NODES(6)
      LNODS(5,1)=NODES(3)
      LNODS(6,1)=NODES(7)
      LNODS(7,1)=NODES(4)
      LNODS(8,1)=NODES(8)
C
C       ******* WRITE COORD ARRAY********
C
       COORD((LNODS(1,1)),1)=X(1)
       COORD((LNODS(1,1)),2)=Y(1)
       COORD((LNODS(1,1)),3)=Z(1)
C
       COORD((LNODS(2,1)),1)=X(5)
       COORD((LNODS(2,1)),2)=Y(5)
       COORD((LNODS(2,1)),3)=Z(5)
C
       COORD((LNODS(3,1)),1)=X(2)
       COORD((LNODS(3,1)),2)=Y(2)
       COORD((LNODS(3,1)),3)=Z(2)
C
       COORD((LNODS(4,1)),1)=X(6)
```

```
      COORD((LNODS(4,1)),2)=Y(6)
      COORD((LNODS(4,1)),3)=Z(6)
C
      COORD((LNODS(5,1)),1)=X(3)
      COORD((LNODS(5,1)),2)=Y(3)
      COORD((LNODS(5,1)),3)=Z(3)
C
      COORD((LNODS(6,1)),1)=X(7)
      COORD((LNODS(6,1)),2)=Y(7)
      COORD((LNODS(6,1)),3)=Z(7)
C
      COORD((LNODS(7,1)),1)=X(4)
      COORD((LNODS(7,1)),2)=Y(4)
      COORD((LNODS(7,1)),3)=Z(4)
C
      COORD((LNODS(8,1)),1)=X(8)
      COORD((LNODS(8,1)),2)=Y(8)
      COORD((LNODS(8,1)),3)=Z(8)
C
      VPROP(1,1) = PROP(1)
      VPROP(2,1) = PROP(2)
      VPROP(3,1) = PROP(3)
      VPROP(4,1) = PRESS(1)
      VPROP(5,1) = RVR(1)
C
C *****  GIVE PREP DATA TO LOOF WHICH CALLS THE OTHER
C          SEMILOOF SUBROUTINES AND RETURNS WITH THE
C          UPPER TRIANGLE STIFFNESS MATRIX ELSTIF 528X1
C          AND ALSO THE LOAD VECTOR ELOAD 32X1.
C
      CALL LOOF(COORD, VPROP, ELSTIF, ELOAD, SVR, LNODS)
C
C
C ******  GIVE ELSTIF 528X1 TO INTEROUT WHICH WILL
C           EXPAND IT TO A 32X32 MATRIX BEFORE
C           PUTTING IT INTO AN ANSYS CORNER, CORNER...
C           MIDSIDE, MIDSIDE.. 48X48 CONFIGURATION.
C
      CALL INTEROUT(ELSTIF, ELOAD, ZS, ZSC)
C
      RETURN
      END
C
C
C
C
C   **************************************************************
C   *  SUBROUTINE  INTEROUT                                      *
C   **************************************************************
C
C
      SUBROUTINE INTEROUT (ELSTIF,ELOAD, ZS, ZSC)
C     ================
C
C     EXPANDS THE UPPER TRIANGLE OF ELSTIF(32*32) TO ZS(48*48)
C
      DOUBLE PRECISION STIF(32,32), ELSTIF(528), ZS(48,48),
     ~ ZS1(48,48)
```

```
      DOUBLE PRECISION ELOAD(32,1), ELOAD1(48,1), ZSC(48,1)
C
      OPEN(UNIT=34,FORM='UNFORMATTED',STATUS='UNKNOWN')
      OPEN(UNIT=39,FORM='FORMATTED',STATUS='UNKNOWN')
C
C
C **** EXPAND ELSTIF 528X1 UPPER K TRIANGLE TO FULL SQUARE
C        MATRIX STIF 32X32 *****
C
      KOUNT = 0
      DO 50 IR=1,32
      DO 51 IC=1,IR
      KOUNT = KOUNT + 1
      STIF(IR,IC) = ELSTIF(KOUNT)
      STIF(IC,IR)=  ELSTIF(KOUNT)
51    CONTINUE
50    CONTINUE
C
C
C ******   WRITE STIF 32X32  TO A FORMATTED FILE *****
C
C      DO 54 IR=1,32
C      WRITE(32,300) IR
C      DO 55 IC=1,32
C      WRITE(32,200) IR, IC, STIF(IR,IC)
C55    CONTINUE
C54    CONTINUE
C100   FORMAT(8X,F16.6)
C200   FORMAT(1X,8H STIF ( ,I2,3H , ,I2,5H ) = ,F16.6)
C300   FORMAT(1X,12H ROW NUMBER ,I2,11H FOLLOWS : )
C
C
C ***** INITIALIZE ZS 48X48 BY PUTTING ZEROES IN THE UNUSED
C         ROWS AND COLUMNS AND ONE'S IN THE RESPECTIVE
C         LEADING DIAGONAL *******
C
      DO 1 I=1,48
      DO 2 J=1,48
      ZS(I,J)=1
      ELOAD1(I,1)=1
2     CONTINUE
1     CONTINUE
C
      DO 5 IROW=4,6
      DO 4 J=1,48
      ZS(IROW,J)=0.0
      ZS(J,IROW)=0.0
      ZS(IROW,IROW)=1.0
      ELOAD1(IROW,1)=0.0
4     CONTINUE
5     CONTINUE
C
      DO 6 IROW=16,18
      DO 7 J=1,48
      ZS(IROW,J)=0.0
      ZS(J,IROW)=0.0
      ZS(IROW,IROW)=1.0
      ELOAD1(IROW,1)=0.0
```

```
7       CONTINUE
6       CONTINUE
C
        DO 9 IROW=28,30
        DO 8 J=1,48
        ZS(IROW,J)=0.0
        ZS(J,IROW)=0.0
        ZS(IROW,IROW)=1.0
        ELOAD1(IROW,1)=0.0
8       CONTINUE
9       CONTINUE
C
        DO 10 IROW=40,42
        DO 11 J=1,48
        ZS(IROW,J)=0.0
        ZS(J,IROW)=0.0
        ZS(IROW,IROW)=1.0
        ELOAD1(IROW,1)=0.0
11      CONTINUE
10      CONTINUE
C
        IROW=12
        DO 12 J=1,48
        ZS(IROW,J)=0.0
        ZS(J,IROW)=0.0
        ZS(IROW,IROW)=1.0
        ELOAD1(IROW,1)=0.0
12      CONTINUE
C
        IROW=24
        DO 13 J=1,48
        ZS(IROW,J)=0.0
        ZS(J,IROW)=0.0
        ZS(IROW,IROW)=1.0
        ELOAD1(IROW,1)=0.0
13      CONTINUE
C
        IROW=36
        DO 14 J=1,48
        ZS(IROW,J)=0.0
        ZS(J,IROW)=0.0
        ZS(IROW,IROW)=1.0
        ELOAD1(IROW,1)=0.0
14      CONTINUE
C
        IROW=48
        DO 15 J=1,48
        ZS(IROW,J)=0.0
        ZS(J,IROW)=0.0
        ZS(IROW,IROW)=1.0
        ELOAD1(IROW,1)=0.0
15      CONTINUE
C
C
C *****    COPY STIF 32X32 TO ZS 48X48 *****
C
        IR=0
        DO 18 I=1,32
```

```
      IC=0
19    IR=IR+1
      IF(IR.EQ.49) GOTO 22
      IF(ZS(IR,1).EQ.0.0) GOTO 19
      DO 20 J=1,32
21    IC=IC+1
      IF(IC.EQ.49) GOTO 18
      IF(ZS(IR,IC).EQ.0.0) GOTO 21
      ZS(IR,IC)=STIF(I,J)
20    CONTINUE
      ELOAD1(IR,1)=ELOAD(I,1)
18    CONTINUE
C
C
C ******  WRITE ZS(48*48) TO A FORMATTED FILE NO. 39  ******
C
22    DO 154 IR=1,48
C      DO 155 IC=1,48
C      WRITE(39,700) IR, IC, ZS(IR,IC)
C155    CONTINUE
154    CONTINUE
C700   FORMAT(1X,8H ZS   ( ,I2,3H , ,I2,5H ) = ,F16.6)
C
C
C ******  WRITE ELOAD TO A FORMATTED FILE NO.55
C
C      DO 25 I=1,48
C      WRITE(55,402) I,ELOAD1(I,1)
C402   FORMAT(1X,9H ELOAD1 (,I2,3H) =,F16.4)
C25    CONTINUE
C
C ****************************************************************
C
C ******  SWAP ROWS AND COLUMNS OF ZS AND ELOAD1 TO
C          SUIT ANSYS'S CORNER, CORNER..., MIDSIDE,
C          MIDSIDE.... NODE CONFIGURATION ****
C
C
C *****  FIRST SWAP ROWS OF ZS INTO ZS1 *****
C
      DO 101 I=1,6
      DO 102 J=1,48
      ZS1(I,J)=ZS(I,J)
      ZSC(I,1)=ELOAD1(I,1)
102     CONTINUE
101     CONTINUE
C
      DO 103 I=7,12
      DO 104 J=1,48
      ZS1(I+18,J)=ZS(I,J)
      ZSC(I+18,1)=ELOAD1(I,1)
104     CONTINUE
103     CONTINUE
C
      DO 105 I=13,18
      DO 106 J=1,48
      ZS1(I-6,J)=ZS(I,J)
      ZSC(I-6,1)=ELOAD1(I,1)
```

```
106     CONTINUE
105     CONTINUE
C
      DO 107 I=19,24
      DO 108 J=1,48
      ZS1(I+12,J)=ZS(I,J)
      ZSC(I+12,1)=ELOAD1(I,1)
108     CONTINUE
107     CONTINUE
C
      DO 109 I=25,30
      DO 110 J=1,48
      ZS1(I-12,J)=ZS(I,J)
      ZSC(I-12,1)=ELOAD1(I,1)
110    CONTINUE
109     CONTINUE
C
      DO 111 I=31,36
      DO 112 J=1,48
      ZS1(I+6,J)=ZS(I,J)
      ZSC(I+6,1)=ELOAD1(I,1)
112     CONTINUE
111     CONTINUE
C
      DO 113 I=37,42
      DO 114 J=1,48
      ZS1(I-18,J)=ZS(I,J)
      ZSC(I-18,1)=ELOAD1(I,1)
114     CONTINUE
113     CONTINUE
C
C
      DO 115 I=43,48
      DO 116 J=1,48
      ZS1(I,J)=ZS(I,J)
      ZSC(I,1)=ELOAD1(I,1)
116     CONTINUE
115     CONTINUE
C
C
C *****   ZERO ZS ******
C
      DO 117 I=1,48
      DO 118 J=1,48
      ZS(I,J)=0
118    CONTINUE
117    CONTINUE
C
C
C ******  SWAP COLUMNS OF ZS1 BACK TO ZS   ******
C
      DO 121 I=1,48
      DO 122 J=1,6
      ZS(I,J)=ZS1(I,J)
122     CONTINUE
121     CONTINUE
C
      DO 123 I=1,48
```

```
      DO 124 J=7,12
      ZS(I,J+18)=ZS1(I,J)
124     CONTINUE
123     CONTINUE
C
      DO 125 I=1,48
      DO 126 J=13,18
      ZS(I,J-6)=ZS1(I,J)
126     CONTINUE
125     CONTINUE
C
      DO 127 I=1,48
      DO 128 J=19,24
      ZS(I,J+12)=ZS1(I,J)
128     CONTINUE
127     CONTINUE
C
      DO 129 I=1,48
      DO 130 J=25,30
      ZS(I,J-12)=ZS1(I,J)
130    CONTINUE
129    CONTINUE
C
      DO 131 I=1,48
      DO 132 J=31,36
      ZS(I,J+6)=ZS1(I,J)
132     CONTINUE
131     CONTINUE
C
      DO 133 I=1,48
      DO 134 J=37,42
      ZS(I,J-18)=ZS1(I,J)
134     CONTINUE
133     CONTINUE
C
C
      DO 135 I=1,48
      DO 136 J=43,48
      ZS(I,J)=ZS1(I,J)
136     CONTINUE
135     CONTINUE
C
C
      RETURN
      END
C
C
C *******************************************************************
```

```
C     ***********************************************************
C     *   STIFFNESS IS A FILE CONTAINING THE SEMILOOF SUBROUTINES  *
C     *   TAKEN FROM TECHNIQUES OF FINITE ELEMENTS BY BRUCE IRONS  *
C     *   ---------------------------------------------------     *
C     *   SUBROUTINES INCLUDED ARE:     BLOCK DATA                 *
C     *                                 LOOF                       *
C     *                                 HALOOF                     *
C     *                                 NFUNC                      *
C     *                                 SCALAR                     *
C     *                                 SFR                        *
C     *                                 VECTOR                     *
C     ***********************************************************
C
C
C ---------------------------------------------------------------
C
C
C      ********************************************************
C      *  BLOCK  DATA                                         *
C      ********************************************************
C
C
      BLOCK DATA
C
C
      DIMENSION COEFA(147), COEFB(100)
      COMMON/COEF/COEF(247)
      COMMON/SYSTEM/NDF(8,2)
      EQUIVALENCE (COEF(1),COEFA(1)), (COEF(148),COEFB(1))
C
C
      DATA COEFA/1., -3., -3., 2., 4., 2., 0., 4., 0., -4., -4., 0., 0.,
     1 -1., 0., 2., 0., 0., 0., 0., 0., 0., 4., 0., 0., 0., -1., 0., 0.,
     2  2., 0., 0., 4., 0., -4., -4., 0.910683603,  1.577350269,
     3 -6.041451884,-6.196152423, 2.464101615, 8.928203230, 1.732050808,
     4 -0.244016936, 0.422649731, 2.041451884, 4.196152423,-4.464101615,
     5 -4.928203230,-1.732050808, 0.333333333,-1.422649731,-2.577350269,
     6 -1.464101615, 5.000000000, 5.464101615, 1.732050808, 0.333333333,
     7 -2.577350269,-1.422649731, 5.464101615, 5.000000000,-1.464101615,
     8 -1.732050808, -.244016936, 2.041451884, 0.422649731,-4.928203230,
     9 -4.464101615, 4.196152423, 1.732050807, 0.910683602,-6.041451884,
     1  1.577350269, 8.928203230, 2.464101615,-6.196152422,-1.732050807,
     2 -1., 6., 6., -6., -6., -6., 0., -.25, 0., 0., .25, .25, .25,
     ~  -.25, -.25, 0.,
     3 .5, 0., -.5, -.5, 0., 0., 0., .5, 0., -.25, 0., 0., .25, -.25,
     ~.25, .25, -.25, 0.,
     4 .5, .5, 0., 0., 0., -.5, -.5, 0., 0., -.25, 0., 0., .25, .25,
     ~ .25, .25, .25, 0.,
     5 .5, 0., .5, -.5, 0., 0., 0., -.5, 0., -.25, 0., 0., .25, -.25,
     ~  .25, -.25, .25/
      DATA COEFB/ 0.,
     6  .5, -.5, 0., 0., 0., -.5, .5, 0., 0., 1., 0., 0.,
     ~  -1., 0., -1., 0., 0., 1,
     ~  0.000000000, 0.216506351, -0.375000000, -0.093750000,
     1  .216506351,  .281250000, -.649519053, 0.375000000, -.324759526,
     2 -.000000000, -.216506351, -.375000000, -.093750000, -.216506351,
     3  .281250000,  .649519053,  .375000000,  .324759526,  .000000000,
     4  .375000000,  .216506351,  .281250000, -.216506351, -.093750000,
```

```
     5 -.375000000, -.649519053, -.324759526, 0.000000000,  .375000000,
     6 -.216506351,  .281250000,  .216506351, -.093750000, -.375000000,
     7  .649519053,  .324759526, -.000000000, -.216506351,  .375000000,
     8 -.093750000,  .216506351,  .281250000,  .649519053, -.375000000,
     9 -.324759526, 0.000000000,  .216506351,  .375000000, -.093750000,
     1 -.216506351,  .281250000, -.649519053, -.375000000,  .324759526,
     2 -.000000000, -.375000000, -.216506351,  .281250000, -.216506351,
     3 -.093750000,  .375000000,  .649519053, -.324759526, -.000000000,
     4 -.375000000,  .216506351,  .281250000,  .216506351, -.093750000,
     5 .375, -.649519053, .324759526, 1., 0., 0., -.75, 0., -.75, 0.,
     ~ 0., 0./
C
C
      DATA NDF/3,5,3,5,3,5,3,5,3,5,3,3,3,0,0,0/
      END
C
C
C
C    ************************************************************
C    *  SUBROUTINE  LOOF                                        *
C    ************************************************************
C
C
      SUBROUTINE LOOF (COORD, VPROP, ELSTIF, ELOAD, SVR, LNODS)
C     ===============
C
C **** DIMENSIONING FILE FOR VARIABLES AND ARRAYS USED BY SEMILOOF ****
C
C
C **** DIMENSIONING FILE FOR VARIABLES AND ARRAYS USED BY SEMILOOF ****
C
C
C **** ANSYS ARRAYS *****
C
      DOUBLE PRECISION ZS(48,48), ZSC(48,1), U(48), RVR(4), SVR(816),
     ~ PROP(3)
C
C **** SEMILOOF ARRAYS ******
C
      INTEGER LNODS(8,1)
      DOUBLE PRECISION COORD(2000,3), VPROP(5,1), ELSTIF(528),
     ~ ELOAD(32,1), ELOAD1(48,1), STRESM(32,6)
C
C
C
      INTEGER I,IGAUS,IGAUSZ,IVAB,IXY,J,JDEL,JVAB,K,KOUNT,LNODZ,
     ~ LNOMAX,LPOP,LVABZ,LVMAX,MAXTRS,N,NBARLO,ND,NDIM,NEL,NELZ,
     ~ NEWRHS,NOD,NODE,NST,NSTRES
C
      DOUBLE PRECISION BEND,DENSTY,GASH,H,PLANE,POIS,PRESS,
     ~ Q,R,S,T,UNIT,WEIGHT,YMOD
      DOUBLE PRECISION B(6,32),BV(192),DB(6),DMOD(6,6),
     ~ XITA(2)
C
C
      COMMON/SHELL/AREA,ELXYZT(9,4),FRAM(3,3),POIN(3),SIDE,THIK,
     ~ WCORN(10,3),WLOOF(10,3),WSHEL(13,45)
C
```

```
     ~ (DB(4),S),(DB(5),T),(DB(6),H)
C
      DIMENSION XGAUS(4,4)
      DATA XGAUS/0., 4*.5, 0., 2*.5, -.577350269, 2*.577350269,
     ~ 3*-.577350269, 2*.577350269/
C
C   ******  INITIALIZE CONSTANTS FOR SEMILOOF ROUTINES ******
C
      NDIM=3
      LPOP=1
      LNODZ=8
      LVABZ=32
      LNOMAX=8
      NELZ=1
      NEL=1
C
C     CONSTANTS FROM STRESS PASS
C
      LVMAX = 32
      MAXTRS = 6
      NEWRHS = 1
      NSTRES = 6
      NBARLO = 4
C
C     INITIALISE ELSTIF AND ELOAD TO ZERO
C
      DO 1131 I=1,528
      ELSTIF(I)=0.0
1131  CONTINUE
C
      DO 1132 I=1,32
      ELOAD(I,1)=0.0
1132  CONTINUE
C
      DO 1133 I=1,48
      ELOAD1(I,1)=0.0
1133  CONTINUE
C
C  *****  END OF INITIALIZATION  *********
C  *****  CODING FROM HERE IS ORIGINAL SEMILOOF *****
C
C
      YMOD=VPROP(1,LPOP)
      POIS=VPROP(2,LPOP)
      DENSTY=VPROP(3,LPOP)
      PRESS=VPROP(4,LPOP)
C      WRITE(6,600) YMOD, POIS, DENSTY, PRESS
C      WRITE(30,600) YMOD, POIS, DENSTY, PRESS
600   FORMAT(/17H YOUNGS MODULUS =,E15.4,10X,17H POISSONS RATIO =,F5.3,
     ~ 10X,10H DENSITY =,E12.6,10X,11H PRESSURE =,E12.6)
C
C
      DO 4 NOD=1,LNODZ
      ELXYZT(NOD,4)= VPROP(5,LPOP)
C
C
      DO 4 ND=1,NDIM
      NODE =IABS(LNODS(NOD,NEL))
```

```
4       ELXYZT(NOD,ND) = COORD(NODE,ND)
C        WRITE(6,602) (LNODS(NOD,NEL),
C       ~ (ELXYZT(NOD,J),J=1,4), NOD=1,LNODZ)
C        WRITE(30,602) (LNODS(NOD,NEL),
C       ~ (ELXYZT(NOD,J),J=1,4), NOD=1,LNODZ)
602     FORMAT(/5H NODE,6X,1HX,12X,1HZ,8X,10H THICKNESS//
      ~ (1X,I3,4F13.7))
C
C
        IGAUSZ=LVABZ/8
C
C *****   SAVE [DB] MATRIX FOR STRESS PASS THROUGH SVR ARRAY ****
C         ZERO COUNTER FOR SVR ITEMS
C
        KOUNT = 0
C
        DO 22 IGAUS=1,IGAUSZ
        DO 8 IXY=1,2
8       XITA(IXY)=XGAUS(IGAUS,IXY+LNODZ-6)
        CALL HALOOF(LNODS,LNODZ,LVABZ,LNOMAX,NEL,NELZ,XITA)
C
C
        DO 10 N=1,36
10      DMOD(N,1) = 0.0
        PLANE = YMOD*THIK*AREA/(1.0-POIS*POIS)
        DMOD(1,1)= PLANE
        DMOD(2,1)= PLANE*POIS
        DMOD(1,2)= PLANE*POIS
        DMOD(2,2)= PLANE
        DMOD(3,3)= PLANE*0.5*(1.0-POIS)
        BEND = PLANE*THIK*THIK/12.0
        DMOD(4,4)=BEND
        DMOD(5,4)=BEND*POIS
        DMOD(4,5)=BEND*POIS
        DMOD(5,5)=BEND
        DMOD(6,6)=BEND*0.5*(1.0-POIS)
C
C
        NST=0
        WEIGHT = AREA*THIK*DENSTY
        DO 20 JVAB=1,LVABZ
        GASH = ELOAD(JVAB,1)
        DO 12 I= 1,3
12      GASH = GASH + WEIGHT*FRAM(3,I)*WSHEL(I,JVAB)
        ELOAD(JVAB,1) = GASH + PRESS*AREA*WSHEL(3,JVAB)
C
C
        B(1,JVAB)= WSHEL(4,JVAB)
        B(2,JVAB)= WSHEL(7,JVAB)
        B(3,JVAB)= WSHEL(5,JVAB) + WSHEL(6,JVAB)
        B(4,JVAB)= WSHEL(10,JVAB)
        B(5,JVAB)= WSHEL(12,JVAB)
        B(6,JVAB)= 2.0*WSHEL(11,JVAB)
C
C
        DO 16 I = 1,6
        GASH = 0.0
        DO 14 K = 1,6
```

```
14     GASH= GASH+DMOD(K,I)*B(K,JVAB)
       DB(I) = GASH
16     STRESM(JVAB,I) = GASH/AREA
C
C
C
       JDEL =1
       DO 18 IVAB = 1,JVAB
       ELSTIF(NST+IVAB) = ELSTIF(NST+IVAB) + P*BV(JDEL) + Q*BV(JDEL+1)
      ~ + R*BV(JDEL+2) + S*BV(JDEL+3) + T*BV(JDEL+4) + H*BV(JDEL+5)
18     JDEL = JDEL + 6
20     NST = NST + JVAB
C
C ***** END OF ORIGINAL SEMILOOF ****
C
C ***   WRITE STRESM MATRIX  **
C
C
C       WRITE(11,363)IGAUS
C363    FORMAT(1X'STRESM MATRIX INTEGRATION POINT ',I)
C       DO 145 M=1,32
C       DO 145 N=1,6
C       WRITE(11,364)M,N,STRESM(M,N)
C364    FORMAT(1X,'STRESM(',I2,',',I2,') = ',F12.5)
C145    CONTINUE
C
C
C      WRITE DB (STRESM) MATRICES FOR EACH BARLOW POINT TO SVR
C
       DO 30 I=1,32
       DO 31 J=1,6
       KOUNT = KOUNT + 1
       SVR(KOUNT) = STRESM(I,J)
31     CONTINUE
30     CONTINUE
C
C       DO 32 I=1,3
C       DO 33 J=1,3
C       KOUNT = KOUNT + 1
C       SVR(KOUNT) = FRAM(I,J)
C33     CONTINUE
C32     CONTINUE
C
C       DO 34 I=1,3
C       KOUNT = KOUNT + 1
C       SVR(KOUNT) = POIN(I)
C34     CONTINUE
C
22     CONTINUE
C
C  ***** WRITE D MATRIX TO SVR  ********
C
       DO 37 I=1,6
       DO 37 J=1,6
       KOUNT = KOUNT + 1
       SVR(KOUNT) = DMOD(I,J)
37     CONTINUE
C
```

```
C
C       WRITE(31,1699) (I,ELSTIF(I),I=1,528)
C      WRITE(33) ELOAD
C1699  FORMAT(1X,1H(,I3,4H) = ,F15.6)
C
C
C       WRITE YOUNGS MODULUS TO SVR
C
      SVR(805)= YMOD
      RETURN
      END
C
C
C
C     *************************************************************
C     *  SUBROUTINE  HALOOF                                       *
C     *************************************************************
C
C
      SUBROUTINE HALOOF(LNODS, LNODZ, LVABZ, LNOMAX, NEL, NELZ, XITA)
C
C
      INTEGER LNODS(LNOMAX,NELZ),NOZPRE,I,NERROR,NSTAGE,LNOD,NX,LIMZ,
     ~ LNODZA,LVABZA,LVABZZ,L,J,LNODZH,KORN,K,JA,NSIDE,LAST,NEXT,MID,
     ~ MIDS,NLOOF,MAYBE,KZ,N,KOL,ND,IZ,NXY,MXY,NZ,LIM,KP,LBIG,NROW
      COMMON/SHELL/AREA, ELXYZT(9,4), FRAM(3,3), POIN(3), SIDE, THIK,
     ~ WCORN(10,3), WLOOF(10,3), WSHEL(13,45)
      DIMENSION GENSID(6,4),XILOOF(9,4),XGAUS(4,4),XYZPRE(8,4)
      DOUBLE PRECISION AREAV(3), FRAME(3,3), POINT(3),
     ~ SHEAR(11,43), SIGT(3), SWOP(6), THIKDD(3,3), TRANS(2,2),
     ~ VLOOF(3,36), XITA(2),XLOCAL(2), XYZDD(3,3), PIVOT,GUSH,GASH,
     ~ GISH,ELMID,FORT,AREASQ,SIDESQ,COSSQ,THIKC,FACT,DETERM,
     ~ PROD,TFIRST,CHANGE
      EQUIVALENCE (T11,TRANS(1,1)), (T12,TRANS(1,2)),
     ~ (T21,TRANS(2,1)), (T22,TRANS(2,2))
      DATA GENSID/1., -1., 0., 3*-.5, 0., 1., -1., 4*1., 0., -1.,
     ~ 4*0., 1., 0., -1., 2*1./ ,XILOOF/.211324866, 2*.788675134,
     ~ .211324866, 2*0.,  .3333333333, 4*0., .211324866, 2*.788675134,
     ~ .211324866, .3333333333, 2*0., -.577350269, .577350269, 2*1.,
     ~ .577350269, -.577350269, 2*-1., 0., 2*-1., -.577350269,
     ~ .577350269, 2*1., .577350269, -.577350269, 0. /,
     ~ XGAUS/0., 4*.5, 0., 2*.5, -.577350269, 2*.577350269,
     ~ 3*-.577350269, 2*.577350269/, XYZPRE/32*0.0/, NOZPRE/0/
C
C
      DO 1 I= 1,585
1     WSHEL(I,1) = 0.0
C
C
      N  ERROR = 1
      IF((LNODZ.NE.6.AND.LNODZ.NE.8).OR .LVABZ.NE.LNODZ*4) GO TO 99
      NSTAGE = 4
      IF (LNODZ.NE.NOZPRE) NSTAGE = 2
      NOZPRE = LNODZ
      DO 2 LNOD =1,LNODZ
      DO 2 NX=1,4
      IF(ELXYZT(LNOD,NX).NE.XYZPRE(LNOD,NX)) NSTAGE=2
2     CONTINUE
```

```
      IF(NSTAGE.EQ.4) GO TO 18
C
C
      LIMZ = (3*LNODZ)/2-1
      LNODZA = LNODZ + 1
      LVABZA = LVABZ + 1
      LVABZZ = LVABZ + LIMZ
      DO 3 L = LNODZA,LIMZ
      DO 3 J = 1,LVABZZ
3     SHEAR(L,J) = 0.0
      DO 5 NX = 1,4
      GASH = 0.0
      LNODZH = LNODZ/2
      DO 4 KORN = 1,LNODZH
      DO 4 K =1,2
4     GASH = GASH
     ~ +8.0*ELXYZT(2*KORN+K-2,NX)/FLOAT(216*K-408-LNODZ*(21*K-41))
5     ELXYZT(9,NX) = GASH
      DO 10 I = 1,LNODZ
      N   ERROR = 2
      IF(ELXYZT(I,4).LE.0.0) GO TO 99
      IF(I.EQ.LNODZ) GO TO 9
      JA = I + 1
      DO 8 J = JA,LNODZ
      N   ERROR = 3
      IF(IABS(LNODS(I,NEL)).EQ.IABS(LNODS(J,NEL))) GO TO 99
      DO 7 K = 1,3
      IF(ELXYZT(I,K).NE.ELXYZT(J,K)) GO TO 8
7     CONTINUE
      N   ERROR = 4
      GO TO 99
8     CONTINUE
9     DO 10 NX = 1,4
      IF(NX.NE.4) ELXYZT(I,NX) = ELXYZT(I,NX) - ELXYZT(9,NX)
10    XYZPRE(I,NX) = ELXYZT(I,NX)
C      WRITE(6,600) ((ELXYZT(I,J),J=1,4), I=1, LNODZ)
C      WRITE(10,600) ((ELXYZT(I,J),J=1,4), I=1, LNODZ)
600   FORMAT(/11X,1HX,12X,1HY,12X,1HZ,12X,1HT/(4X,4F13.7))
C
C
      VLOOF(1,LVABZA) = ELXYZT(9,4)
      DO 12 NSIDE = 1,6
12    SWOP(NSIDE) = 1.0
      LAST = LNODZ - 1
      DO 14 NEXT = 1, LNODZ,2
      MID = LAST +1
      mids=mid/2
      IF(IABS(LNODS(NEXT,NEL)).LT.IABS(LNODS(LAST,NEL)))
     ~ swop(mids) = -1.0
      VLOOF(1,4*LAST-3) = .455341801*ELXYZT(LAST,4)
     ~ + .6666666667*ELXYZT(MID,4) - .122008468*ELXYZT(NEXT,4)
      VLOOF(1,4*MID-3) = -.122008468*ELXYZT(LAST,4)
     ~ + .6666666667*ELXYZT(MID,4) + .455341801*ELXYZT(NEXT,4)
C
C
      GASH = 0.0
      GISH = 0.0
      GUSH = 0.0
```

```
      DO 13 I = 1,3
      ELMID = ELXYZT(MID,I)
      GASH = GASH + (ELXYZT(NEXT,I) - ELMID)**2
      GISH = GISH + (ELXYZT(LAST,I) - ELMID)**2
13    GUSH = GUSH + (ELXYZT(LAST,I) + ELXYZT(NEXT,I)-ELMID-ELMID)**2
      N   ERROR = 5
      IF(DABS(GASH-GISH).GT.0.040*(GASH+GISH)) GO TO 99
      N   ERROR = 6
      IF(GUSH.GT.0.25*(GASH+GISH)) GO TO 99
14    LAST = NEXT
C      WRITE(6,602) SWOP,(VLOOF(1,I), I=1,LVABZA, 4)
C      WRITE(10,602) SWOP,(VLOOF(1,I), I=1,LVABZA, 4)
602   FORMAT(/7H SWOP =,6F6.3//26H THICKNESSES AT LOOF NODES/1X,
     ~ 9F13.8)
C
C
C***ORGANIZE LOOP AROUND LOOF NODES, FOR NSTAGE = 2
C
C     DO 76 NSTAGE = 2,4
15    NLOOF = 0
16    NLOOF = NLOOF +1
C
C      DO 67 NLOOF = 1,LNODZ +1
C      OR DO 67 NLOOF = 1, (3*LNODZ)/2 IF NSTAGE = 3
      DO 17 I =1,2
      IF(NSTAGE.EQ.2. OR .NLOOF.LE.LNODZ)
     ~ XLOCAL(I) = XILOOF(NLOOF,LNODZ+I-6)
C
C***AND ALSO AROUND INTEGRATING POINTS IF NSTAGE=3.
C
      IF (NSTAGE.EQ.3. AND. NLOOF.GT.LNODZ)
     ~ XLOCAL(I) = XGAUS(NLOOF-LNODZ,LNODZ+I-6)
17    CONTINUE
      GO TO 23
C
C***OTHERWISE ORGANISE SINGLE-SHOT OPTION, FOR NSTAGE =4.
C
C***TEST WHETHER INPUT POINT IS A LOOF NODE, PLUS OR MINUS 0.0001.
C
18    DO 19 I = 1,2
19    XLOCAL(I) = XITA(I)
      NLOOF = LNODZA
      DO 22 MAYBE = 1,LNODZ
      DO 20 I =1,2
      FORT = XLOCAL(I)-XILOOF(MAYBE,LNODZ+I-6)
      IF(DABS(FORT).GT.0.0001)  GO TO 22
20    CONTINUE
      NLOOF = MAYBE
22    CONTINUE
C
      IF(NLOOF.LE.LNODZ) WRITE(6,604) NLOOF
      IF(NLOOF.LE.LNODZ) WRITE(10,604) NLOOF
604   FORMAT(/36H INPUT POINT REGOGNIZED AS LOOF NODE,I3)
C
C***CREATE VALUES AND XI, ETA DERIVATIVES OF X,Y,Z,IN XYZDD, T IN THIKDD
C
23    CONTINUE
C      WRITE(6,606)NSTAGE,NLOOF,XLOCAL
```

```
C       WRITE(10,606)NSTAGE,NLOOF,XLOCAL
606    FORMAT(/13H *** NSTAGE =,I2,9H, NLOOF =,I3,6H, XI =,F12.8,
     ~  7H, ETA =,F12.8)
       CALL SFR(LNODZ, NSTAGE,XLOCAL)
       K=0
       DO 27 I =1,3
       DO 26 J = 1,3
       GASH= 0.0
       DO 24 L = 1, LNODZ
24     GASH = GASH +WCORN(L+K,I)*ELXYZT(L,J)
       XYZDD(J,I)= GASH
       IF(NSTAGE.EQ.2) GO TO 26
       GASH = 0.0
       DO 25 L = 1,LNODZA
25     GASH = GASH + WLOOF (L+K,I)*VLOOF(J,4*L-1)
       THIKDD(J,I) = GASH
26     CONTINUE
27     K=1
C       WRITE(6,608) XYZDD
C       WRITE(10,608) XYZDD
608    FORMAT(6H XYZDD/(1X,3F15.10))
C       IF (NSTAGE.EQ.3) WRITE(6,610) THIKDD
C       IF (NSTAGE.EQ.3) WRITE(10,610) THIKDD
610    FORMAT(/7H THIKDD/(1X,3F15.10))
C
C***CREATE VECTOR AREA = VAREA, AT GIVEN POINT XI, ETA.
C
       CALL VECTOR (XYZDD(1,2), XYZDD(1,3), AREAV(1))
       CALL SCALAR (AREAV(1),AREAV(1), AREASQ)
       N  ERROR = 7
       IF(AREASQ.EQ.0.0) GO TO 99
       AREA = SQRT(AREASQ)
C       WRITE(6,612) AREA, AREAV
C       WRITE(10,612) AREA, AREAV
612    FORMAT(/7H AREA =,F13.10,10X,13HAREA VECTOR =, 3F13.10)
C
C***NORMALIZE VECTOR AREA INTO FRAME, COL.3, AS LOCAL UNIT NORMAL Z.
C
C***COLUMN 2 OF FRAME BECOMES UNIT Y AROUND EDGE.
C
       DO  30 I = 1,3
       FRAME (I,3) = AREAV(I)/AREA
       GASH=0.0
       DO 29 J = 1,2
29     GASH = GASH + GENSID ((NLOOF+1)/2,LNODZ+J-6)*XYZDD(I,J+1)
30     FRAME(I,2) = GASH
C
C***NORMALIZE Y, AND IMPLEMENT THICKNESS ETC. INTO VLOOF, FOR INSTAGE = 2
C
       N  ERROR = 8
       CALL SCALAR (FRAME (1,2), FRAME(1,2), SIDESQ)
       IF(SIDESQ.EQ.0.0) GO TO 99
       SIDE = SQRT(SIDESQ)
       DO 31 I = 1,3
       FRAME (I,2) = FRAME(I,2)*SWOP((NLOOF+1)/2)/SIDE
       IF(NSTAGE.NE.2) GO TO 31
       VLOOF (I,4*NLOOF-2) = FRAME(I,2)
       VLOOF (I,4*NLOOF-1) = FRAME(I,3)*VLOOF(1,4*NLOOF-3)
```

```
      VLOOF (I,4*NLOOF)=FRAME(I,3)
31    CONTINUE
C
C***AND COLUMN 1 IS UNIT X, THE OUTWARD IN-PLANE NORMAL
C
      CALL VECTOR(FRAME (1,2), FRAME(1,3), FRAME(1,1))
C      WRITE(6,614) ((FRAME(J,I), I = 1,3), J=1,3)
C      WRITE(10,614) ((FRAME(J,I), I = 1,3), J=1,3)
614   FORMAT (/44H COLS OF FRAME ARE UNIT LOCAL CARTESIAN AXES//,
     ~ (1X,3F13.10))
C
C***CHECK THAT NORMALS ARE REASONABLY PARALLEL, WHILE NSTAGE = 2.
C
      IF(NSTAGE.GT.2) GO TO 35
      IF(NLOOF.EQ.1) GO TO 67
      KZ=4*NLOOF-4
      DO 32 K = 4,KZ,4
      CALL VECTOR (VLOOF(1,4*NLOOF), VLOOF(1,K),POINT(1))
      CALL SCALAR (POINT(1), POINT(1), COSSQ)
      N  ERROR =9
      IF(COSSQ.GT.0.75) GO TO 99
32    CONTINUE
C
C***PLACE CONTRIBUTION OF CENTRAL NODE IN VLOOF (NSTAGE = 2 ONLY)
C
C***COMPLETE NLOOF = 1 TO LNODZ=1 FOR NSTAGE = 2.
C
      IF(NLOOF.LE.LNODZ) GO TO 67
      THIKC = VLOOF (1,LVABZA)
      DO 33 I = 1,3
      DO 33 J = 1,2
33    VLOOF(I,LVABZ+J) = FRAME (I,J)*THIKC
      GO TO 67
C
C***CREATE THE 2X2 JACOBIN MATRIX, AND INVERT IT. (NSTAGE = 3 OR 4)
C
35    DO 36 J = 1,2
      DO 36 I = 1,2
      CALL SCALAR(FRAME(1,I), XYZDD(1,J+1), TRANS(J,I))
36    CONTINUE
C      WRITE(6,616) TRANS
C      WRITE(10,616) TRANS
616   FORMAT(/6H TRANS/(1X,2F13.10))
      GASH = T11
      T11 = T22/AREA
      T22 = GASH/AREA
      T12 = -T12/AREA
      T21 = -T21/AREA
C      WRITE(10,616) TRANS
C      WRITE(6,616) TRANS
C
C***TRANSFORM WCORN AND WLOOF INTO LOCAL X,Y DERIVATIVES.
C
      DO 41 N =1,LNODZA
      DO 41 I =1,2
      GASH = 0.0
      GISH = 0.0
      DO 40 J = 1,2
```

```
      GASH = GASH +TRANS(I,J)*WCORN(N+11,J)
40    GISH = GISH +TRANS(I,J)*WLOOF(N+11,J)
      WCORN(N,I+1) = GASH
41    WLOOF(N,I+1) = GISH
C      WRITE(6,618)
C      WRITE(10,618)
618   FORMAT (1X, 9F13.10)
      DO 42 I = 1,3
C      WRITE(6,620) (WCORN(N,I), N =1,LNODZA)
C      WRITE(10,620) (WCORN(N,I), N =1,LNODZA)
42    CONTINUE
620   FORMAT(1X,9F13.10)
      DO 43 I = 1,3
C      WRITE (6,620) (WLOOF (N,I), N=1,LNODZA)
C      WRITE (10,620) (WLOOF (N,I), N=1,LNODZA)
43    CONTINUE
C
C***PUT THICKNESS AND DERIVATIVES INTO LOCAL COORDINATE SYSTEM.
C
      DO 45 I = 1,3
      DO 44 J = 1,2
      POINT (J)= 0.0
      DO 44 K = 1,2
44    POINT (J) = POINT (J) + TRANS(J,K)*THIKDD(I,K+1)
      DO 45 J =1,2
45    THIKDD(I,J+1)= POINT(J)
      DO 48 J = 1,3
      DO 47 I = 1,3
      CALL SCALAR (THIKDD(1,J), FRAME(1,I), POINT(I))
47    CONTINUE
      DO 48 I = 1,3
48    THIKDD(I,J) = POINT(I)
C      WRITE(6,622) THIKDD
C      WRITE(10,622) THIKDD
622   FORMAT (/17H THICKNESS VECTOR,3F12.7//17H X- DERIVATIVES   ,3F12.7//,
     ~ 17H Y DERIVATIVES   ,3F12.7)
      THIK = THIKDD(3,1)
      N  ERROR = 10
      IF(THIK.LE.0.0) GO TO 99
C
C***FIND THE CHANGE IN X,Y DERIVATIVES ACROSS THICKNESS OF SHELL.
C
      DO 57 LNOD = 1,LNODZA
      IF(NSTAGE.NE.4) GO TO 51
      DO 50 I = 2,3
      GASH = 0.0
      DO 49 J = 1,2
49    GASH = GASH - THIKDD(J,I) *WCORN(LNOD,J+1)
50    POINT(I) = GASH
C
C***CREATE WSHEL = SHAPE FUNCTION ARRAY, DISPLACEMENT TERMS FIRST.
C
51    KORN = (LNOD +1)/2
      DO 54 K = 1,3
      KOL=2*KORN + 3*LNOD +K-5
      IF(LNOD.GT.LNODZ) KOL = 5*LNODZ+2+K
      DO 53 N = 1,3
      FACT = FRAME (K,N)
```

```
      WSHEL(N,KOL) = WCORN(LNOD,1) *FACT
      IF(NSTAGE.EQ.4 .AND .N.EQ.3) FACT=0.0
      DO 53 ND = 2,3
53    WSHEL(N+N+ND,KOL) = WCORN(LNOD,ND)*FACT
      DO 54 N = 1,2
      DO 54 ND = 2,3
      WSHEL(N+7,KOL) = WSHEL(N+7,KOL)
     ~ - THIKDD (ND-1,1) *WSHEL(N+N+ND,KOL)/THIK
      IF (NSTAGE.EQ.4) WSHEL(N+N+ND+6,KOL)=(POINT (ND)* FRAME(K,N)
     ~   + THIKDD(3,ND)*WCORN(LNOD, N+1) * FRAME(K,3))/THIK
54    CONTINUE
C
C***INTRODUCE ROTATION TERMS WITH BENDING ACTION INTO WSHEL.
C
C
      DO 57 L = 1,2
      KOL =(L-1)*4*LNODZ + (2-L)*6*KORN + LNOD
      IF(LNOD.GT.LNODZ) KOL = 5*LNODZ + 3 - L
      DO 56 N = 1,2
      CALL SCALAR(VLOOF(1,4*LNOD+L-4), FRAME(1,N), FACT)
      WSHEL(N+7,KOL) = FACT*WLOOF(LNOD,1)/THIK
      IF(NSTAGE.NE.4) GO TO 56
      DO 55 ND = 2,3
55    WSHEL(N+N+ND+6,KOL) = FACT*WLOOF(LNOD,ND)/THIK
56    CONTINUE
57    CONTINUE
C
C
      IF(LNODZ.EQ.6) GO TO 61
      IZ = 3*NSTAGE + 1
      DO 60 I = 1,IZ
      GASH = 0.0
      DO 59 K = 1,3
59    GASH = GASH + WSHEL(I,42+K)*VLOOF(K,4*LNODZ+4)
60    WSHEL(I,43) = GASH
C      WRITE(6,624) (N, (WSHEL(K,N), K = 1,13), N = 1,LVABZZ)
C      WRITE(10,624) (N, (WSHEL(K,N), K = 1,13), N = 1,LVABZZ)
624   FORMAT(/15H WSHEL ORIGINAL/9X,1HU,8X,1HV,8X,1HW,8X,2HUX,
     ~ 3HVYZ/68X,18HOR UZ+WX OR VZ+WY /(I4,13F9.5))
61    IF(NSTAGE.EQ.4) GO TO 86
C
C
      IF(NLOOF.GT.LNODZ) GO TO 63
      DO 62 I = 1,LVABZZ
      SHEAR(NLOOF,I) = WSHEL(9,I)
      SHEAR(11,I) = SHEAR(11,I) + WSHEL(8,I)*SIDE*THIK*SWOP((NLOOF+1)/2)
62    CONTINUE
      GO TO 67
63    DO 66 KOL = 1,LVABZZ
      DO 66 NXY = 1,2
      GASH = SHEAR(LNODZ+NXY,KOL)
      DO 65 MXY = 1,2
      CALL SCALAR(FRAME(1,MXY), VLOOF(1,4*LNODZ+NXY), FACT)
65    GASH = GASH + WSHEL(MXY+7,KOL)*AREA*THIK*FACT
66    SHEAR(LNODZ+NXY,KOL) = GASH
C
C
67    IF(NLOOF.LE.LNODZ. OR.
```

```
     ~ (NSTAGE.EQ.3 .AND. NLOOF.LT.(3*LNODZ)/2)) GO TO 16
      IF(NSTAGE.NE.2) GO TO 76
C
C
      DO 70 I = 1,3
      GASH = 0.0
      DO 68 N = 3,LVABZ,4
68    GASH = -GASH + VLOOF(I,N)
      SIGT(I) = GASH
C
C
      DO 70 J = 1,3
      GASH = 0.0
      IF(I.EQ.J) GASH = FLOAT(LNODZ)
      DO 69 N = 2,LVABZ,4
69    GASH = GASH - VLOOF(I,N)*VLOOF(J,N)
70    XYZDD(I,J) = GASH
C      WRITE(6,626) SIGT, XYZDD
C      WRITE(10,626) SIGT, XYZDD
626   FORMAT(/24H PLUS-MINUS ERROR VECTOR,3F12.8//
     ~ 25H MATRIX FOR CORRECTING IT,3F12.8,2(/25X,3F12.8))
C
C
      K = 3
      DO 71 I = 1,3
      CALL VECTOR(XYZDD(1,I), XYZDD(1,6-I-K), FRAME(1,K))
71    K = I
      CALL SCALAR(XYZDD(1,1), FRAME(1,1), DETERM)
      DO 73 I=1,3
      CALL SCALAR(FRAME(1,I), SIGT(1), PROD)
73    POINT(I) = PROD/DETERM
C      WRITE(6,628) FRAME, POINT
C      WRITE(10,628) FRAME, POINT
628   FORMAT(/9H ADJUGATE, 3(/1X,3F12.8)//10H SOLUTIONS,3F12.8)
C
C
      FACT = 1.0
      DO 75 N = 2,LVABZ,4
      FACT = -FACT
      CALL SCALAR(POINT(1), VLOOF(1,N), PROD)
      DO 74 I = 1,3
74    VLOOF(I,N+1) = VLOOF(I,N+1) - FACT*(POINT(I)-PROD*VLOOF(I,N))
C
C
      TFIRST=VLOOF(1,N-1)
      CALL VECTOR(VLOOF(1,N), VLOOF(1,N+1), VLOOF(1,N-1))
      DO 75 I = 1,3
75    VLOOF(I,N) = VLOOF(I,N)*TFIRST
      NZ = 4*LNODZA
C      WRITE(6,630) (N, (VLOOF(I,N), I = 1,3),N = 1,NZ)
C      WRITE(10,630) (N, (VLOOF(I,N), I = 1,3),N = 1,NZ)
630   FORMAT(/6H VLOOF/(1X,I3,6X,3F15.10))
      NSTAGE = 3
      GO TO 15
76    CONTINUE
C      WRITE(6,632)
C      WRITE(10,632)
632   FORMAT(/6H SHEAR)
```

```
      DO 77 N = 1,LVABZZ
C      WRITE(6,634) N, (SHEAR(I,N), I = 1,LIMZ)
C      WRITE(10,634) N, (SHEAR(I,N), I = 1,LIMZ)
77    CONTINUE
634   FORMAT(I4,11F10.6)
C
C
      DO 83 LIM = 1,LIMZ
      KP = LVABZ + LIM
      PIVOT = 0.0
      DO 79 L = LIM,LIMZ
      IF(DABS(PIVOT).GT.DABS(SHEAR(L,KP))) GO TO 79
      LBIG = L
      PIVOT = SHEAR(LBIG,KP)
79    CONTINUE
      DO 80 K = 1,LVABZZ
      CHANGE = SHEAR(LBIG,K)
      SHEAR(LBIG,K) = SHEAR(LIM,K)
80    SHEAR(LIM,K) = CHANGE/PIVOT
C
C

      DO 82 NROW = 1,LIMZ
      FACT = SHEAR(NROW,KP)
      IF(NROW.EQ.LIM .OR. FACT.EQ.0.0) GO TO 82
      DO 81 KOL = 1,LVABZZ
81    SHEAR(NROW,KOL) = SHEAR(NROW,KOL) - FACT*SHEAR(LIM,KOL)
82    CONTINUE
83    CONTINUE
C      WRITE(6,636)
C      WRITE(10,636)
636   FORMAT(/22H SHEAR AFTER REDUCTION)
      DO 85 N =1,LVABZZ
C      WRITE(6,634) N, (SHEAR(I,N),I =1,LIMZ)
C      WRITE(10,634) N, (SHEAR(I,N),I =1,LIMZ)
85    CONTINUE
C
      NSTAGE = 4
C      WRITE(6,1600)NSTAGE
C      WRITE(10,1600)NSTAGE
1600  FORMAT(1X,10H NSTAGE = ,I2)
      GO TO 18
C
C
86    DO 88 I = 1,LVABZ
      DO 88 J = 1,13
      GASH = WSHEL(J,I)
      DO 87 K = 1,LIMZ
87    GASH = GASH - WSHEL(J,K+LVABZ)*SHEAR(K,I)
88    WSHEL(J,I) = GASH
C      WRITE(6,638)
C      WRITE(10,638)
638   FORMAT(/18H WSHEL CONSTRAINED)
      DO 89 N = 1,LVABZ
C      WRITE(6,640) N, (WSHEL(J,N), J = 1,13)
C      WRITE(10,640) N, (WSHEL(J,N), J = 1,13)
89    CONTINUE
640   FORMAT(I4,13F9.5)
```

```
C
C
      DO 92 N =8, LVABZ, 8
      IF(SWOP(N/8).EQ.1.0) GO TO 92
      DO 91 J = 1,13
      CHANGE = WSHEL(J,N)
      WSHEL(J,N) = WSHEL(J,N-1)
91    WSHEL(J,N-1) = CHANGE
92    CONTINUE
C      WRITE(6,642)
C      WRITE(10,642)
642   FORMAT(/14H WSHEL SWOPPED)
      DO 94 N = 1,LVABZ
C      WRITE(6,640) N, (WSHEL(J,N), J = 1,13)
C      WRITE(10,640) N, (WSHEL(J,N), J = 1,13)
94    CONTINUE
C
C
C      WRITE(6,644)
C      WRITE(10,644)
644   FORMAT(/30H WSHEL WITH SECOND DERIVATIVES)
      DO 96 N = 1,LVABZ
      WSHEL(10,N) = -WSHEL(10,N)
      WSHEL(11,N) = -0.5*(WSHEL(11,N)+WSHEL(12,N))
      WSHEL(12,N) = -WSHEL(13,N)
C      WRITE(6,640) N, (WSHEL(J,N), J = 1,12)
C      WRITE(10,640) N, (WSHEL(J,N), J = 1,12)
96    CONTINUE
C
C
      AREA = AREA*(FLOAT(LNODZ)-5.6)/2.4
      SIDE = SIDE*FLOAT(LNODZ-4)/4.0
      DO 98 I =1,3
      POIN(I) = XYZDD(I,1) + ELXYZT(9,I)
      DO 98 J = 1,3
98    FRAM(I,J) = FRAME(I,J)
C
      GOTO 123
C
99    WRITE(6,699) NERROR
      WRITE(10,699) NERROR
699   FORMAT(/6H ERROR,I5,18H IN SEGMENT HALOOF)
123   RETURN
      END
C
C
C
C     ****************************************************************
C     *  FUNCTION NFUNC                                              *
C     ****************************************************************
C
C
      FUNCTION NFUNC(I,J)
      NFUNC = J*(J-1)/2 + I
      RETURN
      END
C
C
```

```
C     ************************************************************
C     *  SUBROUTINE  SCALAR                                      *
C     ************************************************************
C
C
      SUBROUTINE SCALAR(U1, V, PROD)
C
      DOUBLE PRECISION U1(3), V(3)
      INTEGER I
      PROD = 0.0
      DO 2 I=1,3
2     PROD = PROD + U1(I)*V(I)
      RETURN
      END
C
C
C     ************************************************************
C     *  SUBROUTINE  SFR                                         *
C     ************************************************************
C
C
      SUBROUTINE SFR(LNODZ, NSTAGE, XLOCAL)
C
C
      COMMON/SHELL/AREA, ELXYZT(9,4), FRAM(3,3), POIN(3), SIDE, THIK,
     ~ WCORN(10,3), WLOOF(10,3), WSHEL(13,45)
      COMMON/COEF/COEF(247)
      DOUBLE PRECISION  XLOCAL(2),ETA,GASH,TERMV,XI
      DIMENSION  TERMV(46)
      INTEGER MD(4),I,IA,IAN,J,K,LNODZ,M,MA,MDEL,MZ,N,N2,N3,NFOIS,
     ~ NFOISZ,NSTAGE,NZ
      DATA MD/8, 43, 90, 171/, TERMV/0.0, 1.0, 44*0.0/
C
C
      XI = XLOCAL(1)
      ETA = XLOCAL(2)
      IF(XI*XI.GT.1.0. OR .ETA*ETA.GT.1.0. OR.
     ~ (LNODZ.EQ.6. AND.(XI.LT.0.0.OR.ETA*(1.000001-XI-ETA).LT.0.0)))
     ~ GO TO 99
      IA = 2
      NZ = (LNODZ+NSTAGE-3)/2
      DO 6 N = 1,NZ
      IAN = IA + N
      N2 = N + 15
      N3 = N + 30
      DO 4 J = IA,IAN
      TERMV(J+N) = TERMV(J)*XI
      TERMV(J +N2) = TERMV(J)*FLOAT(IAN-J)
4     TERMV(J+N3) = TERMV(J-1)*FLOAT(J-IA)
      IA = IAN
6     TERMV(IA+N) = TERMV(IA-1)*ETA
C
C
      DO 8 I = 8,38,15
      IF(LNODZ.EQ.6) TERMV(I) =
     ~ 2.0*(TERMV(I)-TERMV(I+3)) + 3.0*(TERMV(I+1)-TERMV(I+2))
      IF(LNODZ.EQ.8) TERMV(I) = TERMV(I+2)
      IF(LNODZ.EQ.8) TERMV(I+2) = TERMV(I+6)
```

```
8     CONTINUE
C
C
      NFOISZ = (NSTAGE+1)/2
      DO 18 NFOIS = 1, NFOISZ
      NZ = (3*LNODZ)/2 + NFOIS - 4
      IF(NZ.NE.10) GO TO 12
      NZ = 9
      DO 10 I = 10,40,15
10    TERMV(I) = TERMV(I+3) - TERMV(I+5)
12    K = 0
      DO 18 I =1,3
      DO 16 N = 1,NZ
      GASH = 0.0
      MDEL = MD(LNODZ+NFOIS-6) + N*NZ - 15*I
      MA = 16*I-14
      MZ = 15*I+NZ-14
      DO 14 M = MA,MZ
14    GASH = GASH + TERMV(M)*COEF(M+MDEL)
      IF(NFOIS.EQ.1) WCORN(N+K,I) = GASH
      IF(NFOIS.EQ.2) WLOOF(N+K,I) = GASH
16    CONTINUE
18    K = 1
C      WRITE(6,600) XI, ETA, WCORN ,WLOOF
600   FORMAT(/5H XI = F15.10,6X,5HETA =,F15.10//
     - 6H WCORN/3(/1X,10F11.8)//6H WLOOF/3(/1X,10F11.8))
      RETURN
C
C
99    WRITE(6,610) XI, ETA
610   FORMAT(/30H ERROR 11 IN SEGMENT SFR, XI =,F15.9,3X,5HETA = F15.9)
      STOP
      END
C
C
C
C     **************************************************************
C     *  SUBROUTINE  VECTOR                                        *
C     **************************************************************
C
C
      SUBROUTINE VECTOR(U1, V, W)
C
      DOUBLE PRECISION U1(3), V(3), W(3)
      INTEGER I,K
      K = 3
      DO 2 I = 1,3
      W(6-I-K) = U1(K)*V(I) - U1(I)*V(K)
2     K = I
      RETURN
      END
C
C
```

```
C     **************************************************************
C     *   STRESSES IS A FILE CONTAINING BOTH ANSYS AND SEMILOOF      *
C     *   SUBROUTINES NEEDED FOR THE STRESS PASS                     *
C     *   --------------------------------------------------------- *
C     *   SUBROUTINES INCLUDED ARE :  SR100                          *
C     *                               LSTRESS                        *
C     **************************************************************
C
C -----------------------------------------------------------------
C
C     **************************************************************
C     *   SUBROUTINE SR100                                           *
C     **************************************************************
C
C
      SUBROUTINE SR100 (IELNUM,ITYP,KELOUT,ELVOL,KTIK,ZS,ZASS,ZSC)
C     ================
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      EXTERNAL TRACK,GETELD,PUTELD,SRPLT                             EXTERNAL
C
C *****    NEXT FOUR LINES PUTS PLOT FORMAT AS STIF93 *****
C
      INTEGER           IUXX,JELTYP
      REAL              RUXX
      DOUBLE PRECISION DPUXX
      COMMON / COM1 / IUXX(975),RUXX(51),DPUXX(124),JELTYP(20)
C
      INTEGER KDEMO,IPLTAY(6),I
C
C    ***** START STCOM STORAGE ********
      INTEGER   IELNUM,ITYP,KELIN(6),KELOUT(6),NR,KTIK,
     1 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     2 K13,NPRPVL,MATST,K5,K16,IPROP,KCPDS,
     3 K20,KAY,MODE,ISYM,KAHD,IDEBUG,IXXX,
     4 ITYPE,MAT,IELEM,NROW,JTYPE,IPLOT,IPRINT,KTEMTP,KCONCV,KBICNV,
     5 KEYPLS,KEYCRP,KEYSWL,KYSUB(9),K21,NODES(20),    EPAR(50)
      REAL   ERRVAR(5)
      DOUBLE PRECISION                                               DPDCLR
     1 DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     2 TREF,TUNIF,TOFSET,  DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     3 ACEL,OMEGA,CGOMEG,CGLOC,DXXX,
     4 ELMASS,XCENTR,YCENTR,ZCENTR,TFCP,SUBEX,   ERPAR(20),
     5 XYZEQ(20,3),X(20),Y(20),Z(20),   ELVOL
      COMMON /STCOM/ DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     1 TREF,TUNIF,TOFSET,  DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     2 ACEL(3),OMEGA(6),CGOMEG(6),CGLOC(3),          DXXX(16),
     3 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     4 K13,NPRPVL,MATST,K5,K16,IPROP(20),KCPDS,
     5 K20,KAY(10),MODE,ISYM,KAHD,IDEBUG(10),        IXXX(41)
      EQUIVALENCE   (ITYPE,EPAR(1)), (MAT,EPAR(2)), (IELEM,EPAR(5)),
     1 (NROW,EPAR(7)), (JTYPE,EPAR(11)), (IPLOT,EPAR(12)),
     2 (IPRINT,EPAR(13)), (KTEMTP,EPAR(14)), (KCONCV,EPAR(16)),
     4 (KBICNV,EPAR(17)), (KEYPLS,EPAR(18)), (KEYCRP,EPAR(19)),
     5 (KEYSWL,EPAR(20)), (KYSUB(1),EPAR(21)), (K21,EPAR(30)),
     6 (NODES(1),EPAR(31))
      EQUIVALENCE   (ELMASS,ERPAR(1)), (XCENTR,ERPAR(2)),
     1 (YCENTR,ERPAR(3)), (ZCENTR,ERPAR(4)), (TFCP,ERPAR(5)),
```

```
     2 (SUBEX,ERPAR(6))
      EQUIVALENCE (X(1),XYZEQ(1,1)),(Y(1),XYZEQ(1,2)),(Z(1),XYZEQ(1,3))
C     -END STCOM-
C
      DOUBLE PRECISION                                                  *UPD*
     2 AREA,EPORG,
     3 PROP(3),ALEN2,ALENG,DX,DY,DZ,AVETEM,FORCE,EPEL,
     4 EX,ALPX,DENS,POSTD(60)
      INTEGER LP(3)
C
C     THE POSTD ARRAY IS ADDED FOR PLOT FILE ITEM NUMBERING.  THIS IS
C     FOR A ONE TO ONE CORRESPONDANCE BETWEEN THE ITEM NUMBER ON THE PLOT
C     FILE AND THE POSITION IN THE ARRAY.  THE SIZE IS DETERMINED BY
C     THE NUMBER OF ITEMS PUT ON THE PLOT FILE.
C
      DOUBLE PRECISION                                                  *UPD*
     1 DELTEM(1),TEMPER(1),PRESS(1),
     2 RVR(4),SVR(816),
     3 EPTOT,EPTH,SIG,U(240),PLTARY(60),CON
C
      DOUBLE PRECISION ELDISP(32,1), PRIN(2), STRESM(32,6), SG(3)
      DOUBLE PRECISION DIREC(3,2),FRAME(3,3),STRES1(24)
      DOUBLE PRECISION TP(12), AMD(12), BT(12)
      DOUBLE PRECISION PRT(12), PRM(12), PRB(12)
C
      EQUIVALENCE  (PLTARY(1),IPLTAY(1)),   (PLTARY(19),POSTD(1))
C
C     POSTD(1) IS EQUIVALENCED TO
C             PLTARY(11 + NUMBER OF NODES FOR THIS ELEMENT)
C
      EQUIVALENCE (KDEMO,KYSUB(2))
      EQUIVALENCE  (PROP(1),EX), (PROP(2),ALPX), (PROP(3),DENS)
      EQUIVALENCE (RVR(2),FCU)
C
      DOUBLE PRECISION TX, TY, TXY
      DOUBLE PRECISION SXT, SYT, SXYT
      DOUBLE PRECISION SXM, SYM, SXYM
      DOUBLE PRECISION SXB, SYB, SXYB
      DOUBLE PRECISION PT1, PT2
      DOUBLE PRECISION PM1, PM2
      DOUBLE PRECISION PB1, PB2
      DOUBLE PRECISION TXI, TYI, TXYI, TXJ, TYJ, TXYJ
      DOUBLE PRECISION TXK, TYK, TXYK, TXL, TYL, TXYL
      DOUBLE PRECISION SXIT, SYIT, SXYIT, SXJT, SYJT, SXYJT
      DOUBLE PRECISION SXKT, SYKT, SXYKT, SXLT, SYLT, SXYLT
      DOUBLE PRECISION SXIM, SYIM, SXYIM, SXJM, SYJM, SXYJM
      DOUBLE PRECISION SXKM, SYKM, SXYKM, SXLM, SYLM, SXYLM
      DOUBLE PRECISION SXIB, SYIB, SXYIB, SXJB, SYJB, SXYJB
      DOUBLE PRECISION SXKB, SYKB, SXYKB, SXLB, SYLB, SXYLB
      DOUBLE PRECISION AMXI, AMYI, AMXYI, AMXJ, AMYJ, AMXYJ
      DOUBLE PRECISION AMXK, AMYK, AMXYK, AMXL, AMYL, AMXYL
      DOUBLE PRECISION PT1I,PT2I,PT3I,PT1J,PT2J,PT3J
      DOUBLE PRECISION PT1K,PT2K,PT3K,PT1L,PT2L,PT3L
      DOUBLE PRECISION PM1I,PM2I,PM3I,PM1J,PM2J,PM3J
      DOUBLE PRECISION PM1K,PM2K,PM3K,PM1L,PM2L,PM3L
      DOUBLE PRECISION PB1I,PB2I,PB3I,PB1J,PB2J,PB3J
      DOUBLE PRECISION PB1K,PB2K,PB3K,PB1L,PB2L,PB3L
      DOUBLE PRECISION SGT1,SGT2,SGT3
```

```
      DOUBLE PRECISION SGM1,SGM2,SGM3
      DOUBLE PRECISION SGB1,SGB2,SGB3
      DOUBLE PRECISION SGET,SGEM,SGEB
      DOUBLE PRECISION A,B,C,D1,E
      DOUBLE PRECISION CRUSHT, CRUSHM, CRUSHB
      DOUBLE PRECISION CRST,CRSM,CRSB
      DOUBLE PRECISION EQPRT,EQPRM,EQPRB
      DOUBLE PRECISION BETA, TOR2, FC, FBC, FCU
      DOUBLE PRECISION MXTEN, STRAI, ALTN
      DOUBLE PRECISION CKXT,CKXM,CKXB
      DOUBLE PRECISION CKYT,CKYM,CKYB
      DOUBLE PRECISION CXYT,CXYM,CXYB
C
C
C
C
      EQUIVALENCE
     ~ (STRES1(1),TXI),  (STRES1(2),TYI),  (STRES1(3),TXYI),
     ~ (STRES1(4),AMXI),  (STRES1(5),AMYI),  (STRES1(6),AMXYI),
     ~ (STRES1(7),TXJ),  (STRES1(8),TYJ),  (STRES1(9),TXYJ),
     ~ (STRES1(10),AMXJ), (STRES1(11),AMYJ), (STRES1(12),AMXYJ),
     ~ (STRES1(13),TXK), (STRES1(14),TYK), (STRES1(15),TXYK),
     ~ (STRES1(16),AMXK), (STRES1(17),AMYK), (STRES1(18),AMXYK),
     ~ (STRES1(19),TXL), (STRES1(20),TYL), (STRES1(21),TXYL),
     ~ (STRES1(22),AMXL), (STRES1(23),AMYL), (STRES1(24),AMXYL)
      EQUIVALENCE
     ~ (TP(1),SXIT), (TP(2),SYIT), (TP(3),SXYIT),
     ~ (TP(4),SXJT), (TP(5),SYJT), (TP(6),SXYJT),
     ~ (TP(7),SXKT), (TP(8),SYKT), (TP(9),SXYKT),
     ~ (TP(10),SXLT), (TP(11),SYLT), (TP(12),SXYLT)
      EQUIVALENCE
     ~ (AMD(1),SXIM), (AMD(2),SYIM), (AMD(3),SXYIM),
     ~ (AMD(4),SXJM), (AMD(5),SYJM), (AMD(6),SXYJM),
     ~ (AMD(7),SXKM), (AMD(8),SYKM), (AMD(9),SXYKM),
     ~ (AMD(10),SXLM), (AMD(11),SYLM), (AMD(12),SXYLM)
      EQUIVALENCE
     ~ (BT(1),SXIB), (BT(2),SYIB), (BT(3),SXYIB),
     ~ (BT(4),SXJB), (BT(5),SYJB), (BT(6),SXYJB),
     ~ (BT(7),SXKB), (BT(8),SYKB), (BT(9),SXYKB),
     ~ (BT(10),SXLB), (BT(11),SYLB), (BT(12),SXYLB)
      EQUIVALENCE
     ~ (PRT(1),PT1I), (PRT(2),PT2I), (PRT(3),PT3I),
     ~ (PRT(4),PT1J), (PRT(5),PT2J), (PRT(6),PT3J),
     ~ (PRT(7),PT1K), (PRT(8),PT2K), (PRT(9),PT3K),
     ~ (PRT(10),PT1L), (PRT(11),PT2L), (PRT(12),PT3L)
      EQUIVALENCE
     ~ (PRM(1),PM1I), (PRM(2),PM2I), (PRM(3),PM3I),
     ~ (PRM(4),PM1J), (PRM(5),PM2J), (PRM(6),PM3J),
     ~ (PRM(7),PM1K), (PRM(8),PM2K), (PRM(9),PM3K),
     ~ (PRM(10),PM1L), (PRM(11),PM2L), (PRM(12),PM3L)
      EQUIVALENCE
     ~ (PRB(1),PB1I), (PRB(2),PB2I), (PRB(3),PB3I),
     ~ (PRB(4),PB1J), (PRB(5),PB2J), (PRB(6),PB3J),
     ~ (PRB(7),PB1K), (PRB(8),PB2K), (PRB(9),PB3K),
     ~ (PRB(10),PB1L), (PRB(11),PB2L), (PRB(12),PB3L)
C
      OPEN(UNIT=34,FILE='CRUSH.DAT',FORM='FORMATTED',STATUS='NEW')
C
```

```
C
      CALL TRACK (5,'SR100 ')                                            TRACK
      CALL GETELD (IELNUM,ITYP,EPAR(1),ERPAR(1),DELTEM(1),TEMPER(1),    GETELD
     1 PRESS(1),CON,RVR(1),SVR(1),XYZEQ(1,1),U(1))
C
C         *****       STRESSES   *****
C
      CALL LSTRESS (U,SVR,RVR, STRES1,TP,AMD,BT,PRT,PRM,PRB)
C
C
C         ****** WRITE POSTDATA FILE *******
C
 200  IF (IPLOT .NE. 1) GO TO 900
C               *****  NUMBER OF FORCES (LEVEL 1)  *****
      IPLTAY(2) = 6
C               *****  NUMBER OF STRESSES (LEVEL 2)  *****
      IPLTAY(3) = 119
C              *****  NUMBER OF TOTAL SAVED (LEVELS 1, 2, AND 3)  *****
      IPLTAY(4) = 125
C               *****  SAVE GEOMETRY FOR CONTOURS (0,NO  1,YES)  *****
      IPLTAY(6) = 1
C
C        ********* CALCULATE AVERAGE ELEMENT STRESSES *****
C
      TX = ( TXI + TXJ + TXK + TXL ) /4.0
      TY = ( TYI + TYJ + TYK + TYL ) /4.0
      TXY = ( TXYI + TXYJ + TXYK + TXYL ) /4.0
C
      AMX = ( AMXI + AMXJ + AMXK + AMXL ) /4.0
      AMY = ( AMYI + AMYJ + AMYK + AMYL ) /4.0
      AMXY = (AMXYI + AMXYJ + AMXYK + AMXYL ) /4.0
C
      SXT = ( SXIT + SXJT + SXKT + SXLT ) /4.0
      SYT = ( SYIT + SYJT + SYKT + SYLT ) /4.0
      SXYT = ( SXYIT + SXYJT + SXYKT + SXYLT ) /4.0
C
      SXM = ( SXIM + SXJM + SXKM + SXLM ) /4.0
      SYM = ( SYIM + SYJM + SYKM + SYLM ) /4.0
      SXYM = ( SXYIM + SXYJM + SXYKM + SXYLM ) /4.0
C
      SXB = ( SXIB + SXJB + SXKB + SXLB ) /4.0
      SYB = ( SYIB + SYJB + SYKB + SYLB ) /4.0
      SXYB = ( SXYIB + SXYJB + SXYKB + SXYLB ) /4.0
C
C
C ***  CALCULATE CENTROIDAL AVERAGED PRINCIPAL STRESSES ***
C
C  ****  TOP ***
C
      A = (SXT + SYT)*0.5
      B = ((SXT - SYT)**2) + (4*(SXYT**2))
      C = 0.5*(DSQRT(B))
      D1 = A + C
      E = A - C
C
      IF ( D1.GT.E.OR.D1.EQ.E)  GOTO 11
C
      PT1  =  E
```

```
      PT2 =  D1
      GOTO 12
C
11    PT1  = D1
      PT2  = E
C
C
12    CONTINUE
C
C  ****  MIDDLE ***
C
      A = (SXM + SYM)*0.5
      B = ((SXM - SYM)**2) + (4*(SXYM**2))
      C = 0.5*(DSQRT(B))
      D1 = A + C
      E = A - C
C
      IF ( D1.GT.E.OR.D1.EQ.E)  GOTO 13
C
      PM1  =  E
      PM2 =  D1
      GOTO 14
C
13    PM1  = D1
      PM2  = E
C
C
14    CONTINUE
C
C  ****  BOT ***
C
      A = (SXB + SYB)*0.5
      B = ((SXB - SYB)**2) + (4*(SXYB**2))
      C = 0.5*(DSQRT(B))
      D1 = A + C
      E = A - C
C
      IF ( D1.GT.E.OR.D1.EQ.E)  GOTO 15
C
      PB1  =  E
      PB2 =  D1
      GOTO 16
C
15    PB1  = D1
      PB2  = E
C
C
16    CONTINUE
C
c ***  SORT MAX AND MIN CENTROIDAL AVERAGED PRINCIPLES INTO
C      SIG1, SIG2 AND SIG3 ***
C
C
      IF(PT1.GT.0.0.OR.PT1.EQ.0.0) GOTO 50
      SGT1=0.0
      SGT2=PT1
      SGT3=PT2
      GOTO 49
```

```
C
50    SGT1=PT1
      IF(PT2.GT.0.0.OR.PT2.EQ.0.0) GOTO 51
      SGT2=0.0
      SGT3=PT2
      GOTO 49
C
51    SGT2=PT2
      SGT3=0.0
C
49    CONTINUE
C
C
      IF(PM1.GT.0.0.OR.PM1.EQ.0.0) GOTO 53
      SGM1=0.0
      SGM2=PM1
      SGM3=PM2
      GOTO 52
C
53    SGM1=PM1
      IF(PM2.GT.0.0.OR.PM2.EQ.0.0) GOTO 54
      SGM2=0.0
      SGM3=PM2
      GOTO 52
C
54    SGM2=PM2
      SGM3=0.0
C
52    CONTINUE
C
C
      IF(PB1.GT.0.0.OR.PB1.EQ.0.0) GOTO 56
      SGB1=0.0
      SGB2=PB1
      SGB3=PB2
      GOTO 55
C
56    SGB1=PB1
      IF(PB2.GT.0.0.OR.PB2.EQ.0.0) GOTO 57
      SGB2=0.0
      SGB3=PB2
      GOTO 55
C
57    SGB2=PB2
      SGB3=0.0
C
55    CONTINUE
C
C
C ***** CALCULATE VON MISES AT ELEMENT CENTROIDS ***
C
      FACT = 0.707106781
      FACTT= DSQRT(((SGT1-SGT2)**2)+((SGT2-SGT3)**2)+((SGT3-SGT1)**2))
      FACTM= DSQRT(((SGM1-SGM2)**2)+((SGM2-SGM3)**2)+((SGM3-SGM1)**2))
      FACTB= DSQRT(((SGB1-SGB2)**2)+((SGB2-SGB3)**2)+((SGB3-SGB1)**2))
C
      SGET = FACT*FACTT
      SGEM = FACT*FACTM
```

```
      SGEB = FACT*FACTB
C
C
C ***  CALCULATE CRUSHING PARAMETER 'CRUSH'  AT ELEMENT CENTROID ***
C
      EX=SVR(805)

C   ** CALCULATE EQUIVELANT PRESSURE 'EQPRi'
C
      EQPRT = -0.3333333333*(SGT1 + SGT2 + SGT3)
      EQPRM = -0.3333333333*(SGM1 + SGM2 + SGM3)
      EQPRB = -0.3333333333*(SGB1 + SGB2 + SGB3)
C
C  **   CALCULATE CRUSHING PARAMETERS BETA AND TOR2 **
C
      FC=0.78*FCU
      FBC=1.2*FC
C
      BETA = ((FC**2)-(FBC**2)) / ((2*FC)-(4*FBC))
      TOR2 = (FC*FBC*((2*FC)-FBC)) / (3*((2*FBC)-FC))
C
      CRUSHT = ((0.3333333333*(SGET**2))-(BETA*EQPRT))/
     -         ((TOR2)+(BETA*EQPRT))
      CRUSHM = ((0.3333333333*(SGEM**2))-(BETA*EQPRM))/
     -         ((TOR2)+(BETA*EQPRM))
      CRUSHB = ((0.3333333333*(SGEB**2))-(BETA*EQPRB))/
     -         ((TOR2)+(BETA*EQPRB))
C
C
      CRST=0.0
      CRSM=0.0
      CRSB=0.0
C
      IF(CRUSHT.GT.1.0.OR.CRUSHT.EQ.1.0) CRST = CRUSHT
      IF(CRUSHM.GT.1.0.OR.CRUSHM.EQ.1.0) CRSM = CRUSHM
      IF(CRUSHB.GT.1.0.OR.CRUSHB.EQ.1.0) CRSB = CRUSHB
C
C
C  ** WRIT CRUSH PARAMETERS ***
C
C      WRITE(34,2354)BETA,TOR2,EQPRT,CRUSHT,EX
C2354  FORMAT(1X,'BETA = ',F17.5,'TOR2= ',F17.5,'EQPRT= ',F17.5,
C    -           'CRUSHT= ',F17.5,' EX = ',E17.5)
C
C
C **** CALCULATE CRACK PARAMETERS  ****
C
C  ** CALC MAX. TENSILE STRESS AND STRAIN **
C
      MXTEN = 0.1*FC
      STRAI = MXTEN/EX
C
C
      ALTN = 1.1035*MXTEN*DEXP(0.465*STRAI)
C
C
C **** DETERMINE CRACKED ELEMENTS ***
C
```

```
      CKXT=0.0
      CKXM=0.0
      CKXB=0.0
C
      CKYT=0.0
      CKYM=0.0
      CKYB=0.0
C
      CXYT=0.0
      CXYM=0.0
      CXYB=0.0
C
      IF(SXT.GT.0.0) GOTO 61
      IF((SXT+ALTN).LT.0.0) CKXT=(SXT+ALTN)
C
61    IF(SXM.GT.0.0) GOTO 62
      IF((SXM+ALTN).LT.0.0) CKXM=(SXM+ALTN)
C
62    IF(SXB.GT.0.0) GOTO 63
      IF((SXB+ALTN).LT.0.0) CKXB=(SXB+ALTN)
C
63    IF(SYT.GT.0.0) GOTO 64
      IF((SYT+ALTN).LT.0.0) CKYT=(SYT+ALTN)
C
64    IF(SYM.GT.0.0) GOTO 65
      IF((SYM+ALTN).LT.0.0) CKYM=(SYM+ALTN)
C
65    IF(SYB.GT.0.0) GOTO 66
      IF((SYB+ALTN).LT.0.0) CKYB=(SYB+ALTN)
C
66    IF(SXYT.GT.0.0) GOTO 67
      IF((SXYT+ALTN).LT.0.0) CXYT=(SXYT+ALTN)
C
67    IF(SXYM.GT.0.0) GOTO 68
      IF((SXYM+ALTN).LT.0.0) CXYM=(SXYM+ALTN)
C
68    IF(SXYB.GT.0.0) GOTO 69
      IF((SXYB+ALTN).LT.0.0) CXYB=(SXYB+ALTN)
C
69    CONTINUE
C
C  ** WRIT CRACK PARAMETERS ***
C
C      WRITE(34,2355)STRAIT,STRAIM,STRAIB,CRKT,CRKM,CRKB
C2355  FORMAT(1X,'STRAIT = ',F17.5,'STRAIM= ',F17.5,'STRAIB= ',F17.5,
C     -             'CRKT= ',F17.5,'CRKM= ',F17.5,'CRKB= ',F17.5)
C
C      ****** PUT POSTDATA INFORMATION INTO POSTD ******
C


      POSTD(1) = TX
      POSTD(2) = TY
      POSTD(3) = TXY
      POSTD(4) = -AMX
      POSTD(5) = -AMY
      POSTD(6) = -AMXY
C
      POSTD(7)  = SXT
```

```
      POSTD(8)   = SYT
      POSTD(9)   = SXYT
C
      POSTD(10)   = SXM
      POSTD(11)   = SYM
      POSTD(12)   = SXYM
C
      POSTD(13)   = SXB
      POSTD(14)   = SYB
      POSTD(15)   = SXYB
C
      POSTD(16) = SGT1
      POSTD(17) = SGT2
      POSTD(18) = SGT3
C
      POSTD(19) = SGM1
      POSTD(20) = SGM2
      POSTD(21) = SGM3
C
      POSTD(22) = SGB1
      POSTD(23) = SGB2
      POSTD(24) = SGB3
C
      POSTD(25) = SGET
      POSTD(26) = SGEM
      POSTD(27) = SGEB
C
      POSTD(28) = CRST
      POSTD(29) = CRSM
      POSTD(30) = CRSB
C
      POSTD(31) = ALTN
C
      POSTD(32) = CKXT
      POSTD(33) = CKXM
      POSTD(34) = CKXB
C
      POSTD(35) = CKYT
      POSTD(36) = CKYM
      POSTD(37) = CKYB
C
      POSTD(38) = CXYT
      POSTD(39) = CXYM
      POSTD(40) = CXYB
C
C
C
C      IF MORE THAN THE DEFAULT LEVEL OF POSTDATA INFORMATION IS
C      DESIRED, IT SHOULD BE ADDED HERE. KED(K21) WAS SET > 3.
C      IPLTAY(4) MUST BE SET TO THE TOTAL OF ALL LEVELS.
C
C     IF IT IS DESIRED TO PUT THE PLOT FILE IN THE FORMAT OF ANOTHER
C     ELEMENT TYPE(E.G. STIF45), JELTYP(ITYP) MUST BE TEMPORARILY
C     RESET FROM 100 TO 45.  THIS VARIABLE IS NOT IN THE LABELED COMMON
C     REGION STCOM.  RATHER IT IS IN THE LABELED COMMON REGION COM1.
C     THEREFORE, THE FOLLOWING STATEMENTS WOULD NEED TO BE PUT AT THE
C     BEGINNING OF THIS SUBROUTINE(SR100):
C     INTEGER           IUXX,JELTYP
```

```
C     REAL                RUXX
C     DOUBLE PRECISION DPUXX
C     THIS DOUBLE PRECISION SHOULD BE REPLACED WITH A REAL ON
C                   CDC AND CRAY MACHINES.
C     COMMON / COM1 / IUXX(975),RUXX(51),DPUXX(124),JELTYP(20)
C     THE FOLLOWING STATEMENT WOULD NEED TO BE PLACED JUST BEFORE THE
C     CALL TO SRPLT:
C     JELTYP(ITYP) = 45
C     FINALLY, THE FOLLOWING STATEMENT WOULD NEED TO BE PLACED JUST
C     AFTER THE CALL TO SRPLT:
C     JELTYP(ITYP) = 100
C
C     ***** PUT PLTARY INFORMATION ONTO FILE 12 *******
C
      JELTYP(ITYP) = 45
      CALL SRPLT (IELEM,ITYP,NROW,MAT,100,2,U(1),NODES(1),XYZEQ(1,1),          SRPLT
     1 IPLTAY(1),PLTARY(1))
      JELTYP(ITYP) = 100
C
900   CALL PUTELD (IELNUM,EPAR(1),ERPAR(1),CON,SVR(1))                   PUTELD
C     PUTELD RESTORES DATA BACK TO FILE3
      CALL TRACK( 15,'SR100 ')                                           TRACK
      RETURN
      END
C
C
C
C
C    ****************************************************************
C    *   SUBROUTINE LSTRESS                                             *
C    ****************************************************************
C
C
C
      SUBROUTINE LSTRESS (U,SVR,RVR,DBU,TOP,AMID,BOT,SGT,SGM,SGB)
C     =================
C
C     FOUR B MATRICES PER ELEMENT, 1 PER GAUS POINT
C     EACH B MATRIX IS MULTIPLIED SEPERATELY BY D TO FORM DB
C     DB IS THEN STACKED TO FORM DBELEM
C     ARRAY SIZES ARE    D: 6*6    BI=BJ=BK=BL: 6*32  DBELEM : 24*32
C
C
C
      INTEGER I,J,K,L
      DOUBLE PRECISION D(6,6), BI(6,32), BJ(6,32), BK(6,32), BL(6,32)
      DOUBLE PRECISION DBI(6,32), DBJ(6,32), DBK(6,32), DBL(6,32)
      DOUBLE PRECISION DBELEM(24,32), DBU(24), DIR(24)
      DOUBLE PRECISION SVR(816), RVR(4),ELDISP(32,1),U(240)
      DOUBLE PRECISION TOP(12), AMID(12), BOT(12)
      DOUBLE PRECISION PT(8), PM(8), PB(8) ,A ,B, C, D1, E, F
      DOUBLE PRECISION SGT(12), SGM(12), SGB(12)
C
C
C
C    *********   CONVERT ANSYS 48X1 DISPS TO SEMILOOF 32X1  *******
C
      ELDISP(1,1) = U(1)
```

```
      ELDISP(2,1) = U(2)
      ELDISP(3,1) = U(3)
      ELDISP(4,1) = U(25)
      ELDISP(5,1) = U(26)
      ELDISP(6,1) = U(27)
      ELDISP(7,1) = U(28)
      ELDISP(8,1) = U(29)
      ELDISP(9,1) = U(7)
      ELDISP(10,1) = U(8)
      ELDISP(11,1) = U(9)
      ELDISP(12,1) = U(31)
      ELDISP(13,1) = U(32)
      ELDISP(14,1) = U(33)
      ELDISP(15,1) = U(34)
      ELDISP(16,1) = U(35)
      ELDISP(17,1) = U(13)
      ELDISP(18,1) = U(14)
      ELDISP(19,1) = U(15)
      ELDISP(20,1) = U(37)
      ELDISP(21,1) = U(38)
      ELDISP(22,1) = U(39)
      ELDISP(23,1) = U(40)
      ELDISP(24,1) = U(41)
      ELDISP(25,1) = U(19)
      ELDISP(26,1) = U(20)
      ELDISP(27,1) = U(21)
      ELDISP(28,1) = U(43)
      ELDISP(29,1) = U(44)
      ELDISP(30,1) = U(45)
      ELDISP(31,1) = U(46)
      ELDISP(32,1) = U(47)
C
C ***** WRITE DISPLACEMENTS TO FILE42 ****
C
C      DO 21 I=1,32
C      WRITE(35,100)I,ELDISP(I,1)
C100   FORMAT(1X,'ELDISP(',I2,') = ',F17.5)
C21    CONTINUE
C
C
C  *******    READ BACK THE ELASTICITY MATRIX AND THE FOUR STRAIN
C       MATRICES FOR EACH GAUS POINT   ********
C
      KOUNT = 0
C
      DO 1 I=1,32
      DO 1 J=1,6
      KOUNT = KOUNT + 1
      DBI(J,I) = SVR(KOUNT)
C      WRITE(36,200)J,I,DBI(J,I)
C200   FORMAT(1X,'BI(',I2,',',I2,') = ',F17.5)
1     CONTINUE
C
      DO 2 I=1,32
      DO 2 J=1,6
      KOUNT = KOUNT + 1
      DBJ(J,I) = SVR(KOUNT)
C      WRITE(36,300)J,I,DBI(J,I)
```

```
C300   FORMAT(1X,'BJ(',I2,',',I2,') = ',F17.5)
2     CONTINUE
C
      DO 3 I=1,32
      DO 3 J=1,6
      KOUNT = KOUNT + 1
      DBK(J,I) = SVR(KOUNT)
C      WRITE(36,400)J,I,DBI(J,I)
C400   FORMAT(1X,'BK(',I2,',',I2,') = ',F17.5)
3     CONTINUE
C
      DO 4 I=1,32
      DO 4 J=1,6
      KOUNT = KOUNT + 1
      DBL(J,I) = SVR(KOUNT)
C      WRITE(36,500)J,I,DBI(J,I)
C500   FORMAT(1X,'BL(',I2,',',I2,') = ',F17.5)
4     CONTINUE
C
C
      DO 5 I=1,6
      DO 5 J=1,6
      KOUNT = KOUNT + 1
      D(I,J) = SVR(KOUNT)
C      WRITE(37,600)I,J,D(I,J)
C600   FORMAT(1X,'D(',I2,',',I2,') = ',F17.5)
5     CONTINUE
C
C
C *******   NEXT STACK THE FOUR DB MATRICES
C
      DO 11 I=1,6
      DO 11 J=1,32
      DBELEM(I,J) = DBI(I,J)
11    CONTINUE
C
      DO 12 I=1,6
      DO 12 J=1,32
      DBELEM(I+6,J) = DBJ(I,J)
12    CONTINUE
C
      DO 13 I=1,6
      DO 13 J=1,32
      DBELEM(I+12,J) = DBK(I,J)
13    CONTINUE
C
      DO 14 I=1,6
      DO 14 J=1,32
      DBELEM(I+18,J) = DBL(I,J)
14    CONTINUE
C
C
C   *********   MULTIPLY DBELEM*U TO GET STRESSES
C
      DO 15 I=1,24
      DBU(I) = 0.0
15    CONTINUE
      DO 16 I=1,24
```

```
      DO 16 K=1,32
      DBU(I) = DBU(I) + DBELEM(I,K)*ELDISP(K,1)
16    CONTINUE
C
C
C *********   CALCULATE DIRECT STRESSES FROM DIRECT FORCES ***********
C *********   CALCULATE DIRECT STRESSES AT TOP AND BOTTOM FROM MOMENTS  ****
C
C**** DIRECT STRESSES AT MID SURFACE *****
C
      DO 30 I=1,24,6
      DIR(I)   = DBU(I)/RVR(1)
      DIR(I+1) = DBU(I+1)/RVR(1)
      DIR(I+2) = DBU(I+2)/RVR(1)
      DIR(I+3) = DBU(I+3)/(RVR(1)**2)*6
      DIR(I+4) = DBU(I+4)/(RVR(1)**2)*6
      DIR(I+5) = DBU(I+5)/(RVR(1)**2)*6
30    CONTINUE
C
C  ******* CALCULATE TOTAL DIRECT STRESSES (MEMB. + BEND.) AT TOP, MIDDLE
C          AND BOTTOM SURFACES ******
C
      J=-5
      DO 40 I=1,12,3
      J=J+6
      TOP(I)    =  DIR(J) - DIR(J+3)
      TOP(I+1)  =  DIR(J+1) - DIR(J+4)
      TOP(I+2)  =  DIR(J+2) - DIR(J+5)
C
      AMID(I)    =  DIR(J)
      AMID(I+1)  =  DIR(J+1)
      AMID(I+2)  =  DIR(J+2)
C
      BOT(I)    =  DIR(J) + DIR(J+3)
      BOT(I+1)  =  DIR(J+1) + DIR(J+4)
      BOT(I+2)  =  DIR(J+2) + DIR(J+5)
C
40    CONTINUE
C
C
C  ******* CALCULATE PRINCIPAL STRESSES AT TOP MIDDLE AND BOTTOM ****
C
C
C ***  CALCULATE PRINCIPAL STRESSES AT TOP***
C
      J=-1
      DO 32 I=1,12,3
      J=J+2
C
      A = (TOP(I) + TOP(I+1))*0.5
      B = ((TOP(I) - TOP(I+1))**2) + (4*(TOP(I+2)**2))
      C = 0.5*(DSQRT(B))
      D1 = A + C
      E = A - C
C
      IF ( D1.GT.E.OR.D1.EQ.E)  GOTO 31
C
      PT(J)  =  E
```

```
      PT(J+1)  =  D1
      GOTO 32
C
31    PT(J)  = D1
      PT(J+1)  = E
C
32    CONTINUE
C
C ***  CALCULATE PRINCIPAL STRESSES AT MIDDLE ***
C
      J=-1
      DO 34 I=1,12,3
      J=J+2
C
      A = (AMID(I) + AMID(I+1))*0.5
      B = ((AMID(I) - AMID(I+1))**2) + (4*(AMID(I+2)**2))
      C = 0.5*(DSQRT(B))
      D1 = A + C
      E = A - C
C
      IF ( D1.GT.E.OR.D1.EQ.E)  GOTO 35
C
      PM(J)  =  E
      PM(J+1)  =  D1
      GOTO 34
C
35    PM(J)  = D1
      PM(J+1)  = E
C
C
34    CONTINUE
C
C ***  CALCULATE PRINCIPAL STRESSES AT BOTTOM ***
C
      J=-1
      DO 37 I=1,12,3
      J=J+2
C
      A = (BOT(I) + BOT(I+1))*0.5
      B = ((BOT(I) - BOT(I+1))**2) + (4*(BOT(I+2)**2))
      C = 0.5*(DSQRT(B))
      D1 = A + C
      E = A - C
C
      IF ( D1.GT.E.OR.D1.EQ.E)  GOTO 38
C
      PB(J)  =  E
      PB(J+1)  =  D1
      GOTO 37
C
38    PB(J)  = D1
      PB(J+1)  = E
C
C
37    CONTINUE
C
C
c ***  SORT MAX AND MIN PRINCIPLES INTO SIG1, SIG2 AND SIG3 ***
```

```
C
C
      IF(PT(I).GT.0.0.OR.PT(I).EQ.0.0) GOTO 50
      SGT(I)=0.0
      SGT(I+1)=PT(I)
      SGT(I+2)=PT(I+1)
      GOTO 49
C
50    SGT(I)=PT(I)
      IF(PT(I+1).GT.0.0.OR.PT(I+1).EQ.0.0) GOTO 51
      SGT(I+1)=0.0
      SGT(I+2)=PT(I+1)
      GOTO 49
C
51    SGT(I+1)=PT(I+1)
      SGT(I+2)=0.0
C
49    CONTINUE
C
C
      IF(PM(I).GT.0.0.OR.PM(I).EQ.0.0) GOTO 53
      SGM(I)=0.0
      SGM(I+1)=PM(I)
      SGM(I+2)=PM(I+1)
      GOTO 52
C
53    SGM(I)=PM(I)
      IF(PM(I+1).GT.0.0.OR.PM(I+1).EQ.0.0) GOTO 54
      SGM(I+1)=0.0
      SGM(I+2)=PM(I+1)
      GOTO 52
C
54    SGM(I+1)=PM(I+1)
      SGM(I+2)=0.0
C
52    CONTINUE
C
C
      IF(PB(I).GT.0.0.OR.PB(I).EQ.0.0) GOTO 56
      SGB(I)=0.0
      SGB(I+1)=PB(I)
      SGB(I+2)=PB(I+1)
      GOTO 55
C
56    SGB(I)=PB(I)
      IF(PB(I+1).GT.0.0.OR.PB(I+1).EQ.0.0) GOTO 57
      SGB(I+1)=0.0
      SGB(I+2)=PB(I+1)
      GOTO 55
C
57    SGB(I+1)=PB(I+1)
      SGB(I+2)=0.0
C
55    CONTINUE
C
C
      RETURN
      END
```

APPENDIX B

SEMILOOF VERIFICATION ANALYSES

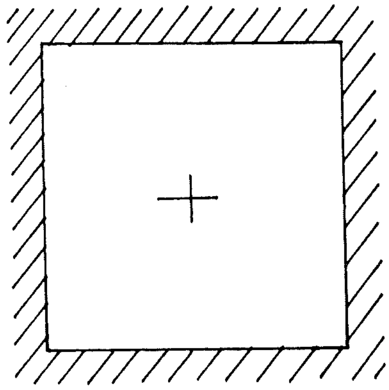
Many simple verification tests were carried out on the SemiLoof shell in ANSYS, and the results of a few of them are reproduced in this Appendix.

The results of a typical test are shown in Table B.1. which show the maximum deflection of the central node in the square plate with fixed edges. The classical result was calculated from the formula in Table B.1 (13). It is interesting to note that SemiLoof converges to the correct theoretical value whereas the ANSYS element, Stif93, does not.

Additional tests, loading the plate in various directions and with varying edge conditions were also conducted with pleasing results, both for displacements and stresses.

The National Association for Finite Element Methods and Standards (NAFEMS (7)), have devised a number of tests, specifically designed to evaluate the performance of thin shell elements. Two of these tests were also conducted with both SemiLoof and Stif93. The test specifications are shown in Tables B.2 and B.3 and the results in Table B.4.

Table B.1. Central deflection of a uniformally loaded square plate with fixed edges.



DATA:

Young's modulus	(E)	=	1000 N/m^2
Poisson's ratio	(v)	=	0.3
Thickness	(t)	=	0.1 m
Breadth = Width	(b)	=	1.0 m
Uniform pressure	(P)	=	10 N/m^2

Central deflection = $0.0138.(P).(b^4) / [E.(t^3)]$ = 0.138 m

Test results:

MESH DENSITY	SEMILOOF	%ERROR	STIF93	%ERRROR
2x2	0.1531	+10.94	0.1913	+38.62
4x4	0.1383	+0.22	0.1639	+18.77
6x6	0.1377	+0.22	0.1644	+19.13
8x8	0.1378	+0.22	0.1643	+19.06

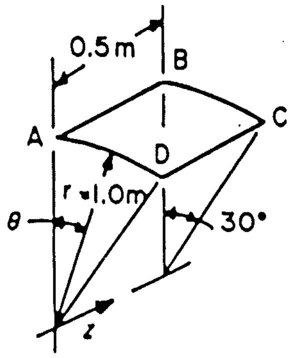
Table B.2 NAFEMS thin shell benchmark specification.

NAFEMS	CYLINDRICAL SHELL PATCH TEST	TEST No LE2	DATE/ISSUE 21-11-86/2

ORIGIN NAFEMS report TSBM

ANALYSIS TYPE Linear elastic

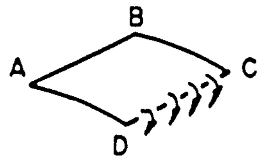
GEOMETRY



Thickness = 0.01m

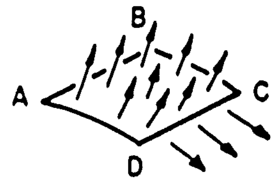
LOADING

Case 1



Uniform normal edge moment, on DC, of 1.0kNm/m

Case 2



Uniform outward normal pressure, at mid-surface ABCD, of 0.6MPa
Tangential outward normal pressure, on edge DC, of 60.0 MPa

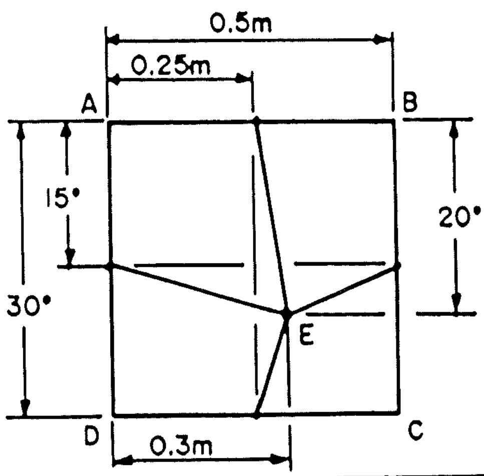
BOUNDARY CONDITIONS Edge AB, all translations and rotations zero
Edge AD and edge BC, symmetry about r θ plane
e.g. z translations and normal rotations all zero

MATERIAL PROPERTIES Isotropic $E = 210 \times 10^3$ MPa, $\nu = 0.3$

ELEMENT TYPES Quadrilateral shells

MESH

(Corner nodes only given)

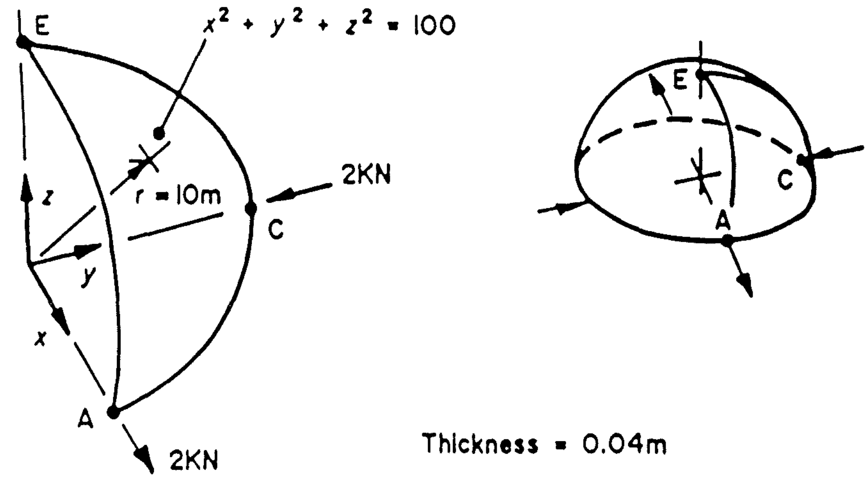


OUTPUT	Outer (convex) surface tangential (θ - θ) stress at point E	TARGET	60.0MPa for both cases

Table B.3 NAFEMS thin shell benchmark specification.

NAFEMS	HEMISPHERE-POINT LOADS	TEST No LE3	DATE/ISSUE 21-11-86/2
ORIGIN	NAFEMS report C1		
ANALYSIS TYPE	Linear elastic		

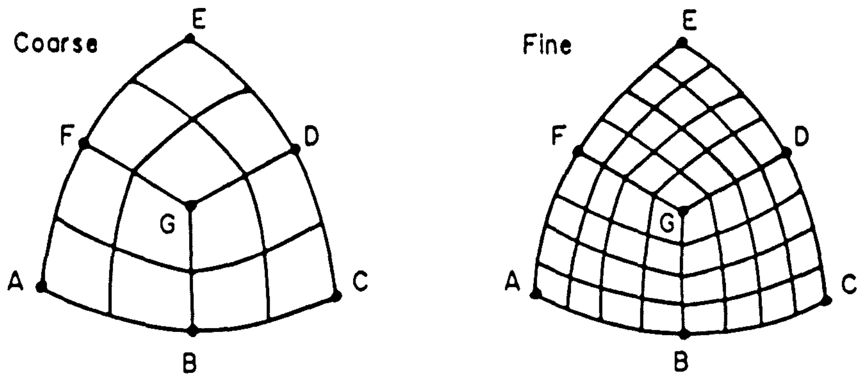
GEOMETRY



Thickness = 0.04m

LOADING	Concentrated radial loads of 2KN outwards at A, inwards at C
BOUNDARY CONDITIONS	Point E, zero z displacement Edge AE, symmetry about zx plane e.g. zero y displacement, zero normal rotation Edge CE, symmetry about yz plane, e.g. zero x displacement, zero normal rotation All other displacements on edge AC are free
MATERIAL PROPERTIES	Isotropic. $E = 68.25 \times 10^3$ MPa, $\nu = 0.3$
ELEMENT TYPES	Quadrilateral shells

MESHES



Equally spaced nodes on AC, CE, EA
Point G at $x = y = z = \frac{10}{\sqrt{3}}$
Equally spaced nodes on BG, DG, FG, (all great circles)

OUTPUT x displacement at point A	TARGET 0.185m

Table B.4 NAFEMS benchmark tests using the SemiLoof shell and ANSYS Stif93.

TEST	TARGET	SEMILOOF % error	STIF93 % error
LE2	60.0 MPa	+8.3	+6.2 (case 2)
LE3	0.185m	-24.4	-25.9 (fine mesh)

APPENDIX C

SUPPORTING PROGRAMS FOR THE SHELTER ANALYSIS

C. 1. Stand-alone Semiloof F.E. Program SEMILOOF.FOR

```
      PROGRAM SEMILOOF
C
      DOUBLE PRECISION ASLOD(32,1), ASTIF(32,32)
      DOUBLE PRECISION FIXED(32)
      DOUBLE PRECISION REACT(32), XDISP(32)
      INTEGER LNODS(8,1),IFPRE(32)
      DOUBLE PRECISION COORD(2000,3), VPROP(5,1), ELSTIF(528)
C
      OPEN(UNIT=2,FILE='OUT',FORM='FORMATTED',STATUS='NEW')
C
C     DEFINE NUMBER OF EQUATIONS
C
      MEQNS=32
      NEQNS=32
C
C
C     ZERO MATRICES
C
      DO 1 I=1,32
      ASLOD(I,1)=0.0
      FIXED(I)=0.0
      IFPRE(I)=0
      REACT(I)=0.0
      XDISP(I)=0.0
      DO 2 J=1,32
      ASTIF(I,J)=0.0
2     CONTINUE
1     CONTINUE
C
C
C     PREPARE DATA TO SEND TO SEMIlOOF
C
C
      LNODS(1,1)=1
      LNODS(2,1)=2
      LNODS(3,1)=3
      LNODS(4,1)=4
      LNODS(5,1)=5
      LNODS(6,1)=6
      LNODS(7,1)=7
      LNODS(8,1)=8
C
C       ******* WRITE COORD ARRAY********
C
       COORD((LNODS(1,1)),1)=0.0
       COORD((LNODS(1,1)),2)=0.0
       COORD((LNODS(1,1)),3)=0.0
C
       COORD((LNODS(2,1)),1)=0.5
       COORD((LNODS(2,1)),2)=0.0
       COORD((LNODS(2,1)),3)=0.0
C
```

```
      COORD((LNODS(3,1)),1)=1.0
      COORD((LNODS(3,1)),2)=0.0
      COORD((LNODS(3,1)),3)=0.0
C
      COORD((LNODS(4,1)),1)=1.0
      COORD((LNODS(4,1)),2)=0.5
      COORD((LNODS(4,1)),3)=0.0
C
      COORD((LNODS(5,1)),1)=1.0
      COORD((LNODS(5,1)),2)=1.0
      COORD((LNODS(5,1)),3)=0.0
C
      COORD((LNODS(6,1)),1)=0.5
      COORD((LNODS(6,1)),2)=1.0
      COORD((LNODS(6,1)),3)=0.0
C
      COORD((LNODS(7,1)),1)=0.0
      COORD((LNODS(7,1)),2)=1.0
      COORD((LNODS(7,1)),3)=0.0
C
      COORD((LNODS(8,1)),1)=0.0
      COORD((LNODS(8,1)),2)=0.5
      COORD((LNODS(8,1)),3)=0.0
C
      VPROP(1,1) = 1000
      VPROP(2,1) = .3
      VPROP(3,1) = 7850
      VPROP(4,1) = 1
      VPROP(5,1) = 0.1
C
C
      CALL LOOF(COORD, VPROP, ELSTIF, ASLOD, LNODS)
C
C
      KOUNT = 0
      DO 50 IR=1,32
      DO 51 IC=1,IR
      KOUNT = KOUNT + 1
      ASTIF(IR,IC) = ELSTIF(KOUNT)
      ASTIF(IC,IR)=  ELSTIF(KOUNT)
51    CONTINUE
50    CONTINUE
C
C
      DO 8 I=1,32
      ASLOD(I,1)=0.0
8     CONTINUE
C
      ASLOD(3,1)=1
      ASLOD(6,1)=2
      ASLOD(11,1)=1
C
C     DEFINE NON-ZERO DISPLACEMENT CONSTRAINTS
C
      FIXED(18)=1E-15
      FIXED(19)=1E-15
      FIXED(21)=1E-15
      FIXED(22)=1E-15
```

```
      FIXED(23)=1E-15
      FIXED(24)=1E-15
      FIXED(25)=1E-15
      FIXED(26)=1E-15
      FIXED(27)=1E-15
C
C     DEFINE FIXITIES
C
      IFPRE(18)=1
      IFPRE(19)=1
      IFPRE(21)=1
      IFPRE(22)=1
      IFPRE(23)=1
      IFPRE(24)=1
      IFPRE(25)=1
      IFPRE(26)=1
      IFPRE(27)=1
C
C
      CALL GREDUC(NEQNS,MEQNS,ASLOD,ASTIF,IFPRE,FIXED)
      CALL BAKSUB(NEQNS,MEQNS,ASLOD,ASTIF,IFPRE,FIXED,
     ~ XDISP,REACT)
C
C
      DO 9 I=1,32
      ASLOD(I,1)=0.0
9     CONTINUE
C
      ASLOD(3,1)=1
      ASLOD(6,1)=2
      ASLOD(11,1)=1
C
      WRITE(2,100) VPROP(1,1),VPROP(2,1)
100   FORMAT(1X,'SINGLE ELEMENT SEMILOOF TEST',//,
     ~ 1X,'YOUNGS MODULUS = ',F10.3,2X,'(N/M2)',
     ~ 12X,'POISSONS RATIO = ',F5.3,//,
     ~ 1X,'D.O.F',3X,'LOAD (N)'3X,'CONSTRAINT',3X,'DISPLACEMENT (M)',
     ~ 3X,'REACTION FORCE (N)')
C
      DO 17 I=1,32
      WRITE(2,200) I,ASLOD(I,1),IFPRE(I),XDISP(I), REACT(I)
17    CONTINUE
200   FORMAT(1X,I2,4X,F10.4,1X,I,4X,F15.4,6X,F15.4)
      STOP
      END
C
C
```

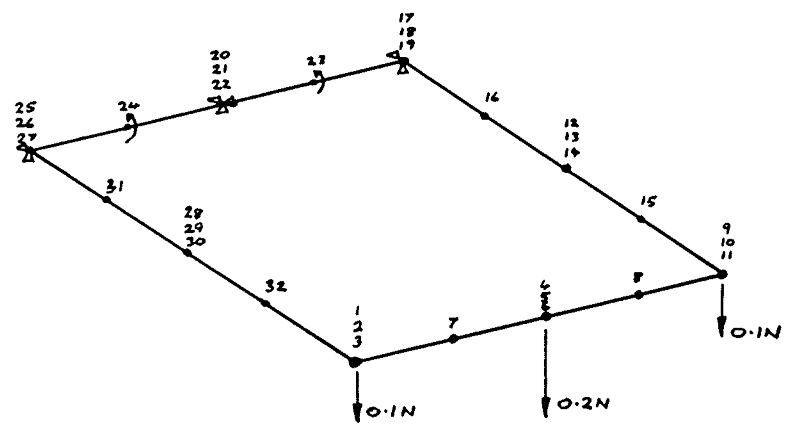
```
      SUBROUTINE GREDUC(NEQNS,MEQNS,ASLOD,ASTIF,
     ~ IFPRE,FIXED)
C
C      GAUSSIAN REDUCTION ROUTINE
C
      DOUBLE PRECISION ASLOD(MEQNS,1),ASTIF(MEQNS,MEQNS)
      DOUBLE PRECISION FIXED(MEQNS)
      INTEGER IFPRE(MEQNS)
      DO 50 IEQNS=1,NEQNS
      IF(IFPRE(IEQNS).EQ.1) GO TO 30
C
C     REDUCE EQUATIONS
C
      PIVOT=ASTIF(IEQNS,IEQNS)
      IF(ABS(PIVOT).LT.1.0E-10) GO TO 60
      IF(IEQNS.EQ.NEQNS) GO TO 50
      IEQN1=IEQNS+1
      DO 20 IROWS=IEQN1,NEQNS
      FACTR=ASTIF(IROWS,IEQNS)/PIVOT
      IF(FACTR.EQ.0.0) GO TO 20
      DO 10 ICOLS=IEQNS,NEQNS
      ASTIF(IROWS,ICOLS)=ASTIF(IROWS,ICOLS)-
     ~ FACTR*ASTIF(IEQNS,ICOLS)
 10   CONTINUE
      ASLOD(IROWS,1)=ASLOD(IROWS,1)-
     ~ FACTR*ASLOD(IEQNS,1)
 20   CONTINUE
      GO TO 50
C
C     ADJUST RHS(LOADS) FOR
C     PRESCRIBED DISPLACEMENTS
C
 30   DO 40 IROWS=IEQNS,NEQNS
      ASLOD(IROWS,1)=ASLOD(IROWS,1)-
     ~ ASTIF(IROWS,IEQNS)*FIXED(IEQNS)
      ASTIF(IROWS,IEQNS)=0.0
 40   CONTINUE
      GO TO 50
 60   WRITE(6,100)
 100  FORMAT(5X,15HINCORRECT PIVOT)
      STOP
 50   CONTINUE
      RETURN
      END
C
```

```
      SUBROUTINE BAKSUB(NEQNS,MEQNS,ASLOD,ASTIF,
     ~ IFPRE, FIXED, XDISP, REACT)
C
C     BACK-SUBSTITUTION ROUTINE
C
      DOUBLE PRECISION ASLOD(MEQNS), ASTIF(MEQNS,MEQNS)
      DOUBLE PRECISION FIXED(MEQNS)
      INTEGER IFPRE(MEQNS)
      DOUBLE PRECISION XDISP(MEQNS), REACT(MEQNS)
      DO 5 IEQNS=1,NEQNS
      REACT(IEQNS)=0.0
5     CONTINUE
      NEQN1=NEQNS+1
      DO 30 IEQNS=1,NEQNS
      NBACK=NEQN1-IEQNS
      PIVOT=ASTIF(NBACK,NBACK)
      RESID=ASLOD(NBACK)
      IF(NBACK.EQ.NEQNS) GO TO 20
      NBAC1=NBACK+1
      DO 10 ICOLS=NBAC1,NEQNS
      RESID=RESID-ASTIF(NBACK,ICOLS)*XDISP(ICOLS)
10    CONTINUE
20    IF(IFPRE(NBACK).EQ.0)
     ~ XDISP(NBACK)=RESID/PIVOT
      IF(IFPRE(NBACK).EQ.1)
     ~ XDISP(NBACK)=FIXED(NBACK)
      IF(IFPRE(NBACK).EQ.1) REACT(NBACK)=-RESID
30    CONTINUE
      RETURN
      END
C
```

SINGLE ELEMENT SEMILOOF TEST - BENDING CASE UZ

YOUNGS MODULUS = 1000.000 (N/M2) POISSONS RATIO = 0.300

D.O.F	LOAD (N)	CONSTRAINT	DISPLACEMENT (M)	REACTION FORCE (N)
1	0.0000	0	0.0000	0.0000
2	0.0000	0	0.0000	0.0000
3	0.1000	0	1.6650	0.0000
4	0.0000	0	0.0000	0.0000
5	0.0000	0	0.0000	0.0000
6	0.2000	0	1.5675	0.0000
7	0.0000	0	-2.4000	0.0000
8	0.0000	0	-2.4000	0.0000
9	0.0000	0	0.0000	0.0000
10	0.0000	0	0.0000	0.0000
11	0.1000	0	1.6650	0.0000
12	0.0000	0	0.0000	0.0000
13	0.0000	0	0.0000	0.0000
14	0.0000	0	0.6712	0.0000
15	0.0000	0	0.1522	0.0000
16	0.0000	0	0.5678	0.0000
17	0.0000	0	0.0000	0.0000
18	0.0000	1	0.0000	0.0000
19	0.0000	1	0.0000	-0.1000
20	0.0000	0	0.0000	0.0000
21	0.0000	1	0.0000	0.0000
22	0.0000	1	0.0000	-0.2000
23	0.0000	1	0.0000	-0.2000
24	0.0000	1	0.0000	-0.2000
25	0.0000	1	0.0000	0.0000
26	0.0000	1	0.0000	0.0000
27	0.0000	1	0.0000	-0.1000
28	0.0000	0	0.0000	0.0000
29	0.0000	0	0.0000	0.0000
30	0.0000	0	0.6712	0.0000
31	0.0000	0	-0.1522	0.0000
32	0.0000	0	-0.5678	0.0000



C.2 SemiLoof mass allocation program MASS.FOR

```
C *******************************************************************
C     THIS PROGRAM READS THE REDUCED MASS DISTRIBUTION TABLE
C     FROM AN ANSYS OUTPUT FILE, AND WRITES THE INFORMATION BACK
C     TO SEPERATE FILES AS MASS ELEMENTS, REAL CONSTANTS AND MASTERS
C
C     INPUT FILE:        TABLE.MASS
C
C     OUTPUT.FILES:      REALS.MASS
C                        ELEMENT.MASS
C                        MASTERS.MASS
C                        SUMMARY.MASS
C *******************************************************************
C
      PROGRAM MASS
C
      DIMENSION NOD(50),AMASS(50),IEL(100)
      CHARACTER*2 DIR(50), ATEMP
C
      OPEN(UNIT=1,FILE='TABLE.MASS',FORM='FORMATTED',STATUS='UNKNOWN')
      OPEN(UNIT=2,FILE='SUMMARY.MASS',FORM='FORMATTED',STATUS='UNKNOWN')
      OPEN(UNIT=3,FILE='REALS.MASS',FORM='FORMATTED',STATUS='UNKNOWN')
      OPEN(UNIT=4,FILE='ELEMENT.MASS',FORM='FORMATTED',STATUS='UNKNOWN')
      OPEN(UNIT=7,FILE='MASTERS.MASS',FORM='FORMATTED',STATUS='UNKNOWN')
C
C     READ DATA
C
C     DEFINE NUMBER OF ITEMS IN TABLE
C
      N=50
C
C     READ TABLE - NODES, DIRECTIONS, MASS
C
      DO 1 I=1,N
      READ(1,100)NOD(I),DIR(I),AMASS(I)
1     CONTINUE
100   FORMAT(8X,I3,2X,A2,5X,F7.2)
C
C     SORT NODES INTO DESCENDING NUMERICAL ORDER
C
4     ISWAP=0
      DO 2 I=1,N
      DO 2 J=I+1,N
      IF(NOD(I).LT.NOD(J).OR.NOD(I).EQ.NOD(J)) GOTO 2
C
      ITEMP=NOD(I)
      NOD(I)=NOD(J)
      NOD(J)=ITEMP
C
      ATEMP=DIR(I)
      DIR(I)=DIR(J)
      DIR(J)=ATEMP
C
      TEMP=AMASS(I)
      AMASS(I)= AMASS(J)
      AMASS(J)=TEMP
```

```
C
      ISWAP=1
2     CONTINUE
      IF(ISWAP.EQ.1) GOTO 4
C
C     PRINT SORTED LINES  TO FILE: SUMMARY.MASS
C
      DO 3 I=1,N
      WRITE(2,200)NOD(I),DIR(I),AMASS(I)
3     CONTINUE
200   FORMAT(1X,I3,3X,A2,3X,F7.2)
C
C     IDENTIFY PAIRS AND WRITE REALS
C
      NR=10
      I=0
6     NR=NR+1
      I=I+1
      R1=0.0
      R2=000.00
      R3=000.00
      IF(DIR(I).EQ.'UX') R1=AMASS(I)
      IF(DIR(I).EQ.'UY') R2=AMASS(I)
      IF(DIR(I).EQ.'UZ') R3=AMASS(I)
      IF(NOD(I).NE.NOD(I+1)) GOTO 5
      I=I+1
      IF(DIR(I).EQ.'UX') R1=AMASS(I)
      IF(DIR(I).EQ.'UY') R2=AMASS(I)
      IF(DIR(I).EQ.'UZ') R3=AMASS(I)
      IF(I.EQ.N-1) GOTO 5
      IF(NOD(I).NE.NOD(I+1)) GOTO 5
      I=I+1
      IF(DIR(I).EQ.'UX') R1=AMASS(I)
      IF(DIR(I).EQ.'UY') R2=AMASS(I)
      IF(DIR(I).EQ.'UZ') R3=AMASS(I)
5     WRITE(3,300)NR,R1,R2,R3
300   FORMAT(1X,'R,',I3,',',F7.2,',',F7.2,',',F7.2)
      IF(I.GT.N) GOTO 7
C
C     NUMBER ELEMENTS
C
      IEL(NR)=NOD(I)
      GOTO 6
C
7     DO 8 I=10,NR
      WRITE(4,400)I,IEL(I)
8     CONTINUE
400   FORMAT(1X,'REAL,',I3,'  $E,',I3)
C
C     DEFINE MASTERS
C
      DO 9 I=1,N
      WRITE(7,500)NOD(I),DIR(I)
9     CONTINUE
500   FORMAT(1X,'M,',I3,',',A2)
C
      STOP
      END
```

Output data from MASS.FOR

```
R, 11,3077.00,   0.00,11497.0
R, 12,1511.10,   0.00,   0.00
R, 13,   0.00,   0.00, 657.02
R, 14,   0.00,   0.00, 747.11
R, 15,   0.00,   0.00, 576.03
R, 16,   0.00,   0.00, 489.63
R, 17,   0.00,   0.00, 500.28
R, 18,   0.00,   0.00,1366.00
R, 19,   0.00,3056.00,   0.00
R, 20,   0.00,2708.20,   0.00
R, 21,2035.70,   0.00,   0.00
R, 22,3305.60,   0.00,   0.00
R, 23,   0.00,   0.00,1891.20
R, 24,   0.00,   0.00, 839.02
R, 25,   0.00,   0.00, 516.67
R, 26,   0.00,   0.00, 875.13
R, 27,8160.50,   0.00,   0.00
R, 28,1065.90,   0.00,   0.00
R, 29, 581.32,   0.00,2447.70
R, 30, 565.34,   0.00,   0.00
R, 31, 661.05,   0.00,   0.00
R, 32,1079.60,   0.00,   0.00
R, 33,   0.00,1156.70,3023.30
R, 34,   0.00,2463.00,4126.60
R, 35,   0.00, 953.79,   0.00
R, 36,3079.20,   0.00,   0.00
R, 37,   0.00,   0.00,1419.50
R, 38,   0.00, 949.65,   0.00
R, 39,   0.00, 769.78,   0.00
R, 40,1399.90,   0.00,   0.00
R, 41,   0.00, 651.15,   0.00
R, 42,   0.00,  73.45,   0.00
R, 43,   0.00, 496.87,   0.00
R, 44,   0.00, 725.10,   0.00
R, 45,   0.00,   0.00,1563.40
R, 46,2992.90,   0.00,   0.00
R, 47, 769.08,   0.00,   0.00
R, 48, 510.90,   0.00,2393.30
R, 49, 460.81,   0.00,   0.00
R, 50, 432.87,   0.00,   0.00
R, 51, 892.47,   0.00,   0.00
R, 52,   0.00,1392.20,   0.00
R, 53,   0.00, 456.07,   0.00
R, 54,   0.00,3783.70,   0.00
R, 55,   0.00,1958.20,   0.00
REAL, 11  $E, 15
REAL, 12  $E, 28
REAL, 13  $E, 33
REAL, 14  $E, 36
REAL, 15  $E, 72
REAL, 16  $E, 79
REAL, 17  $E, 82
REAL, 18  $E, 90
REAL, 19  $E, 99
REAL, 20  $E,173
REAL, 21  $E,205
```

```
REAL, 22  $E,221
REAL, 23  $E,222
REAL, 24  $E,226
REAL, 25  $E,233
REAL, 26  $E,238
REAL, 27  $E,245
REAL, 28  $E,271
REAL, 29  $E,274
REAL, 30  $E,281
REAL, 31  $E,283
REAL, 32  $E,286
REAL, 33  $E,314
REAL, 34  $E,335
REAL, 35  $E,372
REAL, 36  $E,377
REAL, 37  $E,380
REAL, 38  $E,392
REAL, 39  $E,396
REAL, 40  $E,404
REAL, 41  $E,416
REAL, 42  $E,421
REAL, 43  $E,425
REAL, 44  $E,430
REAL, 45  $E,431
REAL, 46  $E,440
REAL, 47  $E,458
REAL, 48  $E,462
REAL, 49  $E,469
REAL, 50  $E,471
REAL, 51  $E,477
REAL, 52  $E,535
REAL, 53  $E,557
REAL, 54  $E,570
REAL, 55  $E,613
M,  15,UZ
M,  15,UX
M,  28,UX
M,  33,UZ
M,  36,UZ
M,  72,UZ
M,  79,UZ
M,  82,UZ
M,  90,UZ
M,  99,UY
M,173,UY
M,205,UX
M,221,UX
M,222,UZ
M,226,UZ
M,233,UZ
M,238,UZ
M,245,UX
M,271,UX
M,274,UZ
M,274,UX
M,281,UX
M,283,UX
M,286,UX
```

M,314,UZ
M,314,UY
M,335,UY
M,335,UZ
M,372,UY
M,377,UX
M,380,UZ
M,392,UY
M,396,UY
M,404,UX
M,416,UY
M,421,UY
M,425,UY
M,430,UY
M,431,UZ
M,440,UX
M,458,UX
M,462,UZ
M,462,UX
M,469,UX
M,471,UX
M,477,UX
M,535,UY
M,557,UY
M,570,UY
M,613,UY

APPENDIX D
ANALYSIS INPUT DATA

D.1 Static analysis

```
/COM SET OVERPRESSURE TO P.  HALF OVERPRESSURE TO Q
P=103350
Q=P/2
/PREP7
/TITLE,HOME OFFICE NUCLEAR SHELTER - STATIC ANALYSIS - STIF100
/VIEW,1,1,1,1
/ANG,1,-120
/SHO,9600,4107,,1
K,1
,2,0.8
,3,2.69
KGEN,2,1,3,1,,2.765
,2,1,3,1,,3.905
,2,1,9,1,,,2.285
,2,1,9,1,,,2.960
/COM LINES
L,1,2,2
,5,4,2
,5,8,2
,8,7,2
,7,4,2
,6,9,2
,10,11,2
,14,13,2
,14,17,2
,17,16,2
,18,15,2
,15,24,2
,23,14,2
,17,26,2
,26,23,2
,27,18,2
,27,24,2
/COM AREAS
A,1,2,5,4
,2,3,6,5
,4,5,8,7
,5,6,9,8
,13,14,11,10
,16,17,14,13
,14,15,12,11
,1,4,13,10
,4,7,16,13
,7,8,17,16
,8,9,18,17
,10,11,2,1
,11,12,3,2
,12,15,6,3
,15,18,9,6
,23,24,15,14
,14,17,26,23
,17,18,27,26
,24,27,18,15
```

```
,4,5,14,13
,5,6,15,14
/COM ELEMENT TYPES
ET,1,93                         * mesh initially with stif93's
/COM MESH                       * to overcome user element meshing limitation
ELSIZ,,4,2
ARSEL,AREA,1,4
AATT,1,1,1
AMES,ALL
ARSEL,AREA,5,7
AATT,1,1,1
AMES,ALL
ARSEL,AREA,8,9
AATT,1,1,1
AMES,ALL
ARSEL,AREA,10,11
AATT,1,1,1
AMES,ALL
ARSEL,AREA,12,13
AATT,1,1,1
AMES,ALL
ARSEL,AREA,14,15
AATT,1,1,1
AMES,ALL
ARSEL,AREA,16,19
AATT,1,1,1
AMES,ALL
ARSEL,AREA,20,21
AATT,2,1,1
AMES,ALL
ARALL
EALL
/COM BLAST DOOR             * change element attributes for steel
NSEL,X,0.4
NRSEL,Y,2.765
NRSEL,Z,0.57,1.1425
ENOD
ERSEL,MAT,2
NELEM
MAT,3
REAL,2
TYPE,1
EMOD,ALL
EALL
NALL
/COM REAL CONSTANTS
R,1,.25                         * thickness
R,2,.015
R,3,.4
/COM MATERIAL PROPERTIES
EX,1,28E9                       * properties for 250mm thick concrete walls
C***DENS,1,2645                 * density included in lumped mass elements
NUXY,1,.15
EX,2,28E9                       * properties for 400mm thick concrete walls
C***DENS,2,2552
NUXY,2,.17
EX,3,210E9                      * properties for 15mm thick steel blast door
C***DENS,3,7850
```

```
NUXY,3,.3
/COM DISPLACEMENT CONSTRAINTS
NSEL,X,0
NRSEL,Y,0
NRSEL,Z,0

D,ALL,UX,,,,,UY,UZ
NSEL,X,0

NRSEL,Y,3.905

NRSEL,Z,0

D,ALL,UX,,,,,UZ
NSEL,X,2.69

NRSEL,Y,0
NRSEL,Z,0

D,ALL,UY,,,,,UZ
NSEL,X,2.69

NRSEL,Y,3.905

NRSEL,Z,0

D,ALL,UZ
NSEL,X,0

NRSEL,Y,0

NRSEL,Z,2.285
      D,ALL,UX,,,,,UY
NSEL,X,0

NRSEL,Y,3.905

NRSEL,Z,2.285
      D,ALL,UX
NSEL,X,2.69

NRSEL,Y,0
NRSEL,Z,2.285
      D,ALL,UY
NALL
EALL
ET,1,100                  * CHANGE ELEMENT TYPE 1 TO SEMILOOF
ET,2,21                   * INTRODUCE ELEMENT TYPE 2, LUMPED MASS ELEMENTS
TYPE,2
/COM
/COM   READ IN LUMPED MASS DATA EXTRACTED BY MASS.FOR
/COM
 R, 11,3077.00,   0.00,11497.0      *  MASS IN X,Y,Z DIRECTIONS
 R, 12,1511.10,   0.00,    0.00
 R, 13,    0.00,   0.00, 657.02
 R, 14,    0.00,   0.00, 747.11
 R, 15,    0.00,   0.00, 576.03
 R, 16,    0.00,   0.00, 489.63
```

```
NUXY,3,.3
/COM DISPLACEMENT CONSTRAINTS
NSEL,X,0
NRSEL,Y,0
NRSEL,Z,0
D,ALL,UX,,,,,UY,UZ
NSEL,X,0
NRSEL,Y,3.905
NRSEL,Z,0
D,ALL,UX,,,,,UZ
NSEL,X,2.69
NRSEL,Y,0
NRSEL,Z,0
D,ALL,UY,,,,,UZ
NSEL,X,2.69
NRSEL,Y,3.905
NRSEL,Z,0
D,ALL,UZ
NSEL,X,0
NRSEL,Y,0
NRSEL,Z,2.285
D,ALL,UX,,,,,UY
NSEL,X,0
NRSEL,Y,3.905
NRSEL,Z,2.285
D,ALL,UX
NSEL,X,2.69
NRSEL,Y,0
NRSEL,Z,2.285
D,ALL,UY
NALL
EALL
ET,1,100                    * CHANGE ELEMENT TYPE 1 TO SEMILOOF
ET,2,21                     * INTRODUCE ELEMENT TYPE 2, LUMPED MASS ELEMENTS
TYPE,2
/COM
/COM   READ IN LUMPED MASS DATA EXTRACTED BY MASS.FOR
/COM
 R, 11,3077.00,   0.00,11497.0     *  MASS IN X,Y,Z DIRECTIONS
 R, 12,1511.10,   0.00,   0.00
 R, 13,   0.00,   0.00, 657.02
 R, 14,   0.00,   0.00, 747.11
 R, 15,   0.00,   0.00, 576.03
 R, 16,   0.00,   0.00, 489.63
 R, 17,   0.00,   0.00, 500.28
 R, 18,   0.00,   0.00,1366.00
 R, 19,   0.00,3056.00,   0.00
 R, 20,   0.00,2708.20,   0.00
 R, 21,2035.70,   0.00,   0.00
 R, 22,3305.60,   0.00,   0.00
 R, 23,   0.00,   0.00,1891.20
 R, 24,   0.00,   0.00, 839.02
 R, 25,   0.00,   0.00, 516.67
 R, 26,   0.00,   0.00, 875.13
 R, 27,8160.50,   0.00,   0.00
 R, 28,1065.90,   0.00,   0.00
 R, 29, 581.32,   0.00,2447.70
 R, 30, 565.34,   0.00,   0.00
```

```
R, 31, 661.05,    0.00,    0.00
R, 32,1079.60,    0.00,    0.00
R, 33,    0.00,1156.70,3023.30
R, 34,    0.00,2463.00,4126.60
R, 35,    0.00, 953.79,    0.00
R, 36,3079.20,    0.00,    0.00
R, 37,    0.00,    0.00,1419.50
R, 38,    0.00, 949.65,    0.00
R, 39,    0.00, 769.78,    0.00
R, 40,1399.90,    0.00,    0.00
R, 41,    0.00, 651.15,    0.00
R, 42,    0.00,  73.45,    0.00
R, 43,    0.00, 496.87,    0.00
R, 44,    0.00, 725.10,    0.00
R, 45,    0.00,    0.00,1563.40
R, 46,2992.90,    0.00,    0.00
R, 47, 769.08,    0.00,    0.00
R, 48, 510.90,    0.00,2393.30
R, 49, 460.81,    0.00,    0.00
R, 50, 432.87,    0.00,    0.00
R, 51, 892.47,    0.00,    0.00
R, 52,    0.00,1392.20,    0.00
R, 53,    0.00, 456.07,    0.00
R, 54,    0.00,3783.70,    0.00
R, 55,    0.00,1958.20,    0.00
REAL, 11  $E, 15                     * DEFINE LUMPED MASS ELEMENTS
REAL, 12  $E, 28
REAL, 13  $E, 33
REAL, 14  $E, 36
REAL, 15  $E, 72
REAL, 16  $E, 79
REAL, 17  $E, 82
REAL, 18  $E, 90
REAL, 19  $E, 99
REAL, 20  $E,173
REAL, 21  $E,205
REAL, 22  $E,221
REAL, 23  $E,222
REAL, 24  $E,226
REAL, 25  $E,233
REAL, 26  $E,238
REAL, 27  $E,245
REAL, 28  $E,271
REAL, 29  $E,274
REAL, 30  $E,281
REAL, 31  $E,283
REAL, 32  $E,286
REAL, 33  $E,314
REAL, 34  $E,335
REAL, 35  $E,372
REAL, 36  $E,377
REAL, 37  $E,380
REAL, 38  $E,392
REAL, 39  $E,396
REAL, 40  $E,404
REAL, 41  $E,416
REAL, 42  $E,421
REAL, 43  $E,425
```

```
 REAL, 44  $E,430
 REAL, 45  $E,431
 REAL, 46  $E,440
 REAL, 47  $E,458
 REAL, 48  $E,462
 REAL, 49  $E,469
 REAL, 50  $E,471
 REAL, 51  $E,477
 REAL, 52  $E,535
 REAL, 53  $E,557
 REAL, 54  $E,570
 REAL, 55  $E,613
/COM  NO MASTERS REQUIRED FOR STATIC
WSORT,X                           * OPTIMISE WAVE FRONT
WSORT,Y
KAN,0                             * STATIC ANALYSIS TYPE
/COM
/COM DEFINE PRESSURE LOADS
/COM
ITER,1,1,1
TIME,0
LWRI
PRST,-1
/COM  ROOF + BLAST WALL
ARSEL,AREA,5
ARASEL,AREA,7
ARASEL,AREA,20,21
EAREA
NELEM
PSF,ALL,,,P
/COM  DWELLING WALLS
ARSEL,AREA,8
ARASEL,AREA,12,14
EAREA
NELEM
PSF,ALL,,,Q
/COM STAIRWELL EXTERNAL WALLS
ARSEL,AREA,9,11
ARASEL,AREA,15
EAREA
NELEM
PSF,ALL,,,-Q
/COM  STAIRWELL FLOOR
ARSEL,AREA,3,4
EAREA
NELEM
PSF,ALL,,,-P
EALL
NALL
AFWRI
FINI
/EXE                          * SUBMIT ANALYSIS FILE FOR EXECUTION
/INP,27
FINI
/EOF
```

D.2. Modal analysis

```
/PREP7
/TITLE,HOME OFFICE NUCLEAR SHELTER - MODAL ANALYSIS - STIF100
/VIEW,1,1,1,1
/ANG,1,-120
/SHO,9600,4107,,1
K,1
,2,0.8
,3,2.69
KGEN,2,1,3,1,,2.765
,2,1,3,1,,3.905
,2,1,9,1,,,2.285
,2,1,9,1,,,2.960
/COM LINES
L,1,2,2
,5,4,2
,5,8,2
,8,7,2
,7,4,2
,6,9,2
,10,11,2
,14,13,2
,14,17,2
,17,16,2
,18,15,2
,15,24,2
,23,14,2
,17,26,2
,26,23,2
,27,18,2
,27,24,2
/COM AREAS
A,1,2,5,4
,2,3,6,5
,4,5,8,7
,5,6,9,8
,13,14,11,10
,16,17,14,13
,14,15,12,11
,1,4,13,10
,4,7,16,13
,7,8,17,16
,8,9,18,17
,10,11,2,1
,11,12,3,2
,12,15,6,3
,15,18,9,6
,23,24,15,14
,14,17,26,23
,17,18,27,26
,24,27,18,15
,4,5,14,13
,5,6,15,14
/COM ELEMENT TYPES
ET,1,93                    * mesh initially with stif93's
/COM MESH                  * to overcome user element meshing limitation
ELSIZ,,4,2
```

```
ARSEL,AREA,1,4
AATT,1,1,1
AMES,ALL
ARSEL,AREA,5,7
AATT,1,1,1
AMES,ALL
ARSEL,AREA,8,9
AATT,1,1,1
AMES,ALL
ARSEL,AREA,10,11
AATT,1,1,1
AMES,ALL
ARSEL,AREA,12,13
AATT,1,1,1
AMES,ALL
ARSEL,AREA,14,15
AATT,1,1,1
AMES,ALL
ARSEL,AREA,16,19
AATT,1,1,1
AMES,ALL
ARSEL,AREA,20,21
AATT,2,1,1
AMES,ALL
ARALL
EALL
/COM BLAST DOOR            * change element attributes for steel
NSEL,X,0.4
NRSEL,Y,2.765
NRSEL,Z,0.57,1.1425
ENOD
ERSEL,MAT,2
NELEM
MAT,3
REAL,2
TYPE,1
EMOD,ALL
EALL
NALL
/COM REAL CONSTANTS
R,1,.25                         * thickness
R,2,.015
R,3,.4
/COM MATERIAL PROPERTIES
EX,1,28E9                     * properties for 250mm thick concrete walls
C***DENS,1,2645               * density included in lumped mass elements
NUXY,1,.15
EX,2,28E9                     * properties for 400mm thick concrete walls
C***DENS,2,2552
NUXY,2,.17
EX,3,210E9                    * properties for 15mm thick steel blast door
C***DENS,3,7850
NUXY,3,.3
/COM DISPLACEMENT CONSTRAINTS
NSEL,X,0
NRSEL,Y,0
NRSEL,Z,0
```

```
D,ALL,UX,,,,,UY,UZ
NSEL,X,0
NRSEL,Y,3.905
NRSEL,Z,0
D,ALL,UX,,,,,UZ
NSEL,X,2.69
NRSEL,Y,0
NRSEL,Z,0
D,ALL,UY,,,,,UZ
NSEL,X,2.69
NRSEL,Y,3.905
NRSEL,Z,0
D,ALL,UZ
NSEL,X,0
NRSEL,Y,0
NRSEL,Z,2.285
D,ALL,UX,,,,,UY
NSEL,X,0
NRSEL,Y,3.905
NRSEL,Z,2.285
D,ALL,UX
NSEL,X,2.69
NRSEL,Y,0
NRSEL,Z,2.285
D,ALL,UY
NALL
EALL
ET,1,100                 * CHANGE ELEMENT TYPE 1 TO SEMILOOF
ET,2,21                  * INTRODUCE ELEMENT TYPE 2, LUMPED MASS ELEMENTS
TYPE,2
/COM
/COM   READ IN LUMPED MASS DATA EXTRACTED BY MASS.FOR
/COM
 R, 11,3077.00,   0.00,11497.0      *  MASS IN X,Y,Z DIRECTIONS
 R, 12,1511.10,   0.00,   0.00
 R, 13,    0.00,   0.00, 657.02
 R, 14,    0.00,   0.00, 747.11
 R, 15,    0.00,   0.00, 576.03
 R, 16,    0.00,   0.00, 489.63
 R, 17,    0.00,   0.00, 500.28
 R, 18,    0.00,   0.00,1366.00
 R, 19,    0.00,3056.00,   0.00
 R, 20,    0.00,2708.20,   0.00
 R, 21,2035.70,   0.00,   0.00
 R, 22,3305.60,   0.00,   0.00
 R, 23,    0.00,   0.00,1891.20
 R, 24,    0.00,   0.00, 839.02
 R, 25,    0.00,   0.00, 516.67
 R, 26,    0.00,   0.00, 875.13
 R, 27,8160.50,   0.00,   0.00
 R, 28,1065.90,   0.00,   0.00
 R, 29, 581.32,   0.00,2447.70
 R, 30, 565.34,   0.00,   0.00
 R, 31, 661.05,   0.00,   0.00
 R, 32,1079.60,   0.00,   0.00
 R, 33,    0.00,1156.70,3023.30
 R, 34,    0.00,2463.00,4126.60
 R, 35,    0.00, 953.79,   0.00
```

```
R, 36,3079.20,   0.00,   0.00
R, 37,    0.00,   0.00,1419.50
R, 38,    0.00, 949.65,   0.00
R, 39,    0.00, 769.78,   0.00
R, 40,1399.90,   0.00,   0.00
R, 41,    0.00, 651.15,   0.00
R, 42,    0.00,  73.45,   0.00
R, 43,    0.00, 496.87,   0.00
R, 44,    0.00, 725.10,   0.00
R, 45,    0.00,   0.00,1563.40
R, 46,2992.90,   0.00,   0.00
R, 47, 769.08,   0.00,   0.00
R, 48, 510.90,   0.00,2393.30
R, 49, 460.81,   0.00,   0.00
R, 50, 432.87,   0.00,   0.00
R, 51, 892.47,   0.00,   0.00
R, 52,    0.00,1392.20,   0.00
R, 53,    0.00, 456.07,   0.00
R, 54,    0.00,3783.70,   0.00
R, 55,    0.00,1958.20,   0.00
REAL, 11  $E, 15                         * DEFINE LUMPED MASS ELEMENTS
REAL, 12  $E, 28
REAL, 13  $E, 33
REAL, 14  $E, 36
REAL, 15  $E, 72
REAL, 16  $E, 79
REAL, 17  $E, 82
REAL, 18  $E, 90
REAL, 19  $E, 99
REAL, 20  $E,173
REAL, 21  $E,205
REAL, 22  $E,221
REAL, 23  $E,222
REAL, 24  $E,226
REAL, 25  $E,233
REAL, 26  $E,238
REAL, 27  $E,245
REAL, 28  $E,271
REAL, 29  $E,274
REAL, 30  $E,281
REAL, 31  $E,283
REAL, 32  $E,286
REAL, 33  $E,314
REAL, 34  $E,335
REAL, 35  $E,372
REAL, 36  $E,377
REAL, 37  $E,380
REAL, 38  $E,392
REAL, 39  $E,396
REAL, 40  $E,404
REAL, 41  $E,416
REAL, 42  $E,421
REAL, 43  $E,425
REAL, 44  $E,430
REAL, 45  $E,431
REAL, 46  $E,440
REAL, 47  $E,458
REAL, 48  $E,462
```

```
 REAL, 49  $E,469
 REAL, 50  $E,471
 REAL, 51  $E,477
 REAL, 52  $E,535
 REAL, 53  $E,557
 REAL, 54  $E,570
 REAL, 55  $E,613
/COM  DEFINE MASTERS REQUIRED FOR MODAL
 M, 15,UZ
 M, 15,UX
 M, 28,UX
 M, 33,UZ
 M, 36,UZ
 M, 72,UZ
 M, 79,UZ
 M, 82,UZ
 M, 90,UZ
 M, 99,UY
 M,173,UY
 M,205,UX
 M,221,UX
 M,222,UZ
 M,226,UZ
 M,233,UZ
 M,238,UZ
 M,245,UX
 M,271,UX
 M,274,UZ
 M,274,UX
 M,281,UX
 M,283,UX
 M,286,UX
 M,314,UZ
 M,314,UY
 M,335,UY
 M,335,UZ
 M,372,UY
 M,377,UX
 M,380,UZ
 M,392,UY
 M,396,UY
 M,404,UX
 M,416,UY
 M,421,UY
 M,425,UY
 M,430,UY
 M,431,UZ
 M,440,UX
 M,458,UX
 M,462,UZ
 M,462,UX
 M,469,UX
 M,471,UX
 M,477,UX
 M,535,UY
 M,557,UY
 M,570,UY
 M,613,UY
```

```
WSORT,X                             * OPTIMISE WAVE FRONT
WSORT,Y
KAN,2                               * MODAL ANALYSIS TYPE
KAY,7,10                           * EXTRACT 10 NATURAL FREQUENCIES
KAY,2,10                           * EXPAND  10 MODE SHAPES
ITER,1,1,1
AFWRI
FINI
/EXE                                * SUBMIT ANALYSIS FILE FOR EXECUTION
/INP,27
FINI
/EOF
```

D.3. Transient analysis

```
/COM SET OVERPRESSURE TO P.  HALF OVERPRESSURE TO Q
P=103350
Q=P/2
/PREP7
/TITLE,HOME OFFICE NUCLEAR SHELTER - TRANSIENT ANALYSIS - STIF100
/VIEW,1,1,1,1
/ANG,1,-120
/SHO,9600,4107,,1
K,1
,2,0.8
,3,2.69
KGEN,2,1,3,1,,2.765
,2,1,3,1,,3.905
,2,1,9,1,,,2.285
,2,1,9,1,,,2.960
/COM LINES
L,1,2,2
,5,4,2
,5,8,2
,8,7,2
,7,4,2
,6,9,2
,10,11,2
,14,13,2
,14,17,2
,17,16,2
,18,15,2
,15,24,2
,23,14,2
,17,26,2
,26,23,2
,27,18,2
,27,24,2
/COM AREAS
A,1,2,5,4
,2,3,6,5
,4,5,8,7
,5,6,9,8
,13,14,11,10
,16,17,14,13
,14,15,12,11
,1,4,13,10
,4,7,16,13
,7,8,17,16
,8,9,18,17
,10,11,2,1
,11,12,3,2
,12,15,6,3
,15,18,9,6
,23,24,15,14
,14,17,26,23
,17,18,27,26
,24,27,18,15
,4,5,14,13
,5,6,15,14
/COM ELEMENT TYPES
```

```
ET,1,93                          * mesh initially with stif93's
/COM MESH                        * to overcome user element meshing limitation
ELSIZ,,4,2
ARSEL,AREA,1,4
AATT,1,1,1
AMES,ALL
ARSEL,AREA,5,7
AATT,1,1,1
AMES,ALL
ARSEL,AREA,8,9
AATT,1,1,1
AMES,ALL
ARSEL,AREA,10,11
AATT,1,1,1
AMES,ALL
ARSEL,AREA,12,13
AATT,1,1,1
AMES,ALL
ARSEL,AREA,14,15
AATT,1,1,1
AMES,ALL
ARSEL,AREA,16,19
AATT,1,1,1
AMES,ALL
ARSEL,AREA,20,21
AATT,2,1,1
AMES,ALL
ARALL
EALL
/COM BLAST DOOR              * change element attributes for steel
NSEL,X,0.4
NRSEL,Y,2.765
NRSEL,Z,0.57,1.1425
ENOD
ERSEL,MAT,2
NELEM
MAT,3
REAL,2
TYPE,1
EMOD,ALL
EALL
NALL
/COM REAL CONSTANTS
R,1,.25                           * thickness
R,2,.015
R,3,.4
/COM MATERIAL PROPERTIES
EX,1,28E9                        * properties for 250mm thick concrete walls
C***DENS,1,2645                  * density included in lumped mass elements
NUXY,1,.15
EX,2,28E9                        * properties for 400mm thick concrete walls
C***DENS,2,2552
NUXY,2,.17
EX,3,210E9                       * properties for 15mm thick steel blast door
C***DENS,3,7850
NUXY,3,.3
/COM DISPLACEMENT CONSTRAINTS
NSEL,X,0
```

```
NRSEL,Y,0
NRSEL,Z,0
D,ALL,UX,,,,,UY,UZ
NSEL,X,0
NRSEL,Y,3.905
NRSEL,Z,0
D,ALL,UX,,,,,UZ
NSEL,X,2.69
NRSEL,Y,0
NRSEL,Z,0
D,ALL,UY,,,,,UZ
NSEL,X,2.69
NRSEL,Y,3.905
NRSEL,Z,0
D,ALL,UZ
NSEL,X,0
NRSEL,Y,0
NRSEL,Z,2.285
D,ALL,UX,,,,,UY
NSEL,X,0
NRSEL,Y,3.905
NRSEL,Z,2.285
D,ALL,UX
NSEL,X,2.69
NRSEL,Y,0
NRSEL,Z,2.285
D,ALL,UY
NALL
EALL
ET,1,100                    * CHANGE ELEMENT TYPE 1 TO SEMILOOF
ET,2,21                     * INTRODUCE ELEMENT TYPE 2, LUMPED MASS ELEMENTS
TYPE,2
/COM
/COM   READ IN LUMPED MASS DATA EXTRACTED BY MASS.FOR
/COM
 R, 11,3077.00,   0.00,11497.0      *  MASS IN X,Y,Z DIRECTIONS
 R, 12,1511.10,   0.00,   0.00
 R, 13,    0.00,   0.00, 657.02
 R, 14,    0.00,   0.00, 747.11
 R, 15,    0.00,   0.00, 576.03
 R, 16,    0.00,   0.00, 489.63
 R, 17,    0.00,   0.00, 500.28
 R, 18,    0.00,   0.00,1366.00
 R, 19,    0.00,3056.00,   0.00
 R, 20,    0.00,2708.20,   0.00
 R, 21,2035.70,   0.00,   0.00
 R, 22,3305.60,   0.00,   0.00
 R, 23,    0.00,   0.00,1891.20
 R, 24,    0.00,   0.00, 839.02
 R, 25,    0.00,   0.00, 516.67
 R, 26,    0.00,   0.00, 875.13
 R, 27,8160.50,   0.00,   0.00
 R, 28,1065.90,   0.00,   0.00
 R, 29, 581.32,   0.00,2447.70
 R, 30, 565.34,   0.00,   0.00
 R, 31, 661.05,   0.00,   0.00
 R, 32,1079.60,   0.00,   0.00
 R, 33,    0.00,1156.70,3023.30
```

```
R, 34,    0.00,2463.00,4126.60
R, 35,    0.00, 953.79,    0.00
R, 36,3079.20,    0.00,    0.00
R, 37,    0.00,    0.00,1419.50
R, 38,    0.00, 949.65,    0.00
R, 39,    0.00, 769.78,    0.00
R, 40,1399.90,    0.00,    0.00
R, 41,    0.00, 651.15,    0.00
R, 42,    0.00,  73.45,    0.00
R, 43,    0.00, 496.87,    0.00
R, 44,    0.00, 725.10,    0.00
R, 45,    0.00,    0.00,1563.40
R, 46,2992.90,    0.00,    0.00
R, 47, 769.08,    0.00,    0.00
R, 48, 510.90,    0.00,2393.30
R, 49, 460.81,    0.00,    0.00
R, 50, 432.87,    0.00,    0.00
R, 51, 892.47,    0.00,    0.00
R, 52,    0.00,1392.20,    0.00
R, 53,    0.00, 456.07,    0.00
R, 54,    0.00,3783.70,    0.00
R, 55,    0.00,1958.20,    0.00
REAL, 11  $E, 15                  * DEFINE LUMPED MASS ELEMENTS
REAL, 12  $E, 28
REAL, 13  $E, 33
REAL, 14  $E, 36
REAL, 15  $E, 72
REAL, 16  $E, 79
REAL, 17  $E, 82
REAL, 18  $E, 90
REAL, 19  $E, 99
REAL, 20  $E,173
REAL, 21  $E,205
REAL, 22  $E,221
REAL, 23  $E,222
REAL, 24  $E,226
REAL, 25  $E,233
REAL, 26  $E,238
REAL, 27  $E,245
REAL, 28  $E,271
REAL, 29  $E,274
REAL, 30  $E,281
REAL, 31  $E,283
REAL, 32  $E,286
REAL, 33  $E,314
REAL, 34  $E,335
REAL, 35  $E,372
REAL, 36  $E,377
REAL, 37  $E,380
REAL, 38  $E,392
REAL, 39  $E,396
REAL, 40  $E,404
REAL, 41  $E,416
REAL, 42  $E,421
REAL, 43  $E,425
REAL, 44  $E,430
REAL, 45  $E,431
REAL, 46  $E,440
```

```
 REAL, 47  $E,458
 REAL, 48  $E,462
 REAL, 49  $E,469
 REAL, 50  $E,471
 REAL, 51  $E,477
 REAL, 52  $E,535
 REAL, 53  $E,557
 REAL, 54  $E,570
 REAL, 55  $E,613
/COM  NO MASTERS REQUIRED FOR KAN4
WSORT,X                             * OPTIMISE WAVE FRONT
WSORT,Y
KAN,4                               * TRANSIENT ANALYSIS TYPE
ALPHAD,32.32                        * MASS DAMPING FACTOR
BETAD,0.00007                       * STIFFNESS DAMPING FACTOR
/COM DEFINE PRESSURE LOADS
/COM
/COM  LOAD CASE 1  TIME = 0  INITIALISE WITH ZERO LOAD
/COM
ITER,1,1,1
TIME,0
LWRI
/COM
/COM  LOAD CASE 2  TIME = 0 TO 0.0005 SECS
/COM
KBC,0
PRST,-1
ITER,1,1,1
TIME,.0005
/COM  ROOF + BLAST WALL
ARSEL,AREA,5
ARASEL,AREA,7
ARASEL,AREA,20,21
EAREA
NELEM
PSF,ALL,,,P
/COM  DWELLING WALLS
ARSEL,AREA,8
ARASEL,AREA,12,14
EAREA
NELEM
PSF,ALL,,,Q
/COM STAIRWELL EXTERNAL WALLS
ARSEL,AREA,9,11
ARASEL,AREA,15
EAREA
NELEM
PSF,ALL,,,-Q
/COM  STAIRWELL FLOOR
ARSEL,AREA,3,4
EAREA
NELEM
PSF,ALL,,,-P
EALL
NALL
LWRIT
/COM
/COM LOAD CASE 3 TIME = 0.0005 TO 0.1 SECS
```

```
/COM
TIME,.1
ITER,-199,1,1
A=0.02            *  TIME/TOTAL TIME = .1/5
B=EXP(-A)
C=1-A
P=P*C*B            * CALCULATE PRESSURE AFTER .1 SECS
Q=P/2              * P = P(1-t/T) e-(t/T)    EQN.2.4
/COM  ROOF + BLAST WALL
ARSEL,AREA,5
ARASEL,AREA,7
ARASEL,AREA,20,21
EAREA
NELEM
PSF,ALL,,,P
/COM  DWELLING WALLS
ARSEL,AREA,8
ARASEL,AREA,12,14
EAREA
NELEM
PSF,ALL,,,Q
/COM STAIRWELL EXTERNAL WALLS
ARSEL,AREA,9,11
ARASEL,AREA,15
EAREA
NELEM
PSF,ALL,,,-Q
/COM  STAIRWELL FLOOR
ARSEL,AREA,3,4
EAREA
NELEM
PSF,ALL,,,-P
EALL
NALL
LWRIT
AFWRI
FINI
/EXE                         * SUBMIT ANALYSIS FILE FOR EXECUTION
/INP,27
FINI
/EOF
```

APPENDIX E
HOME OFFICE SHELTER CONSTRUCTION DRAWINGS

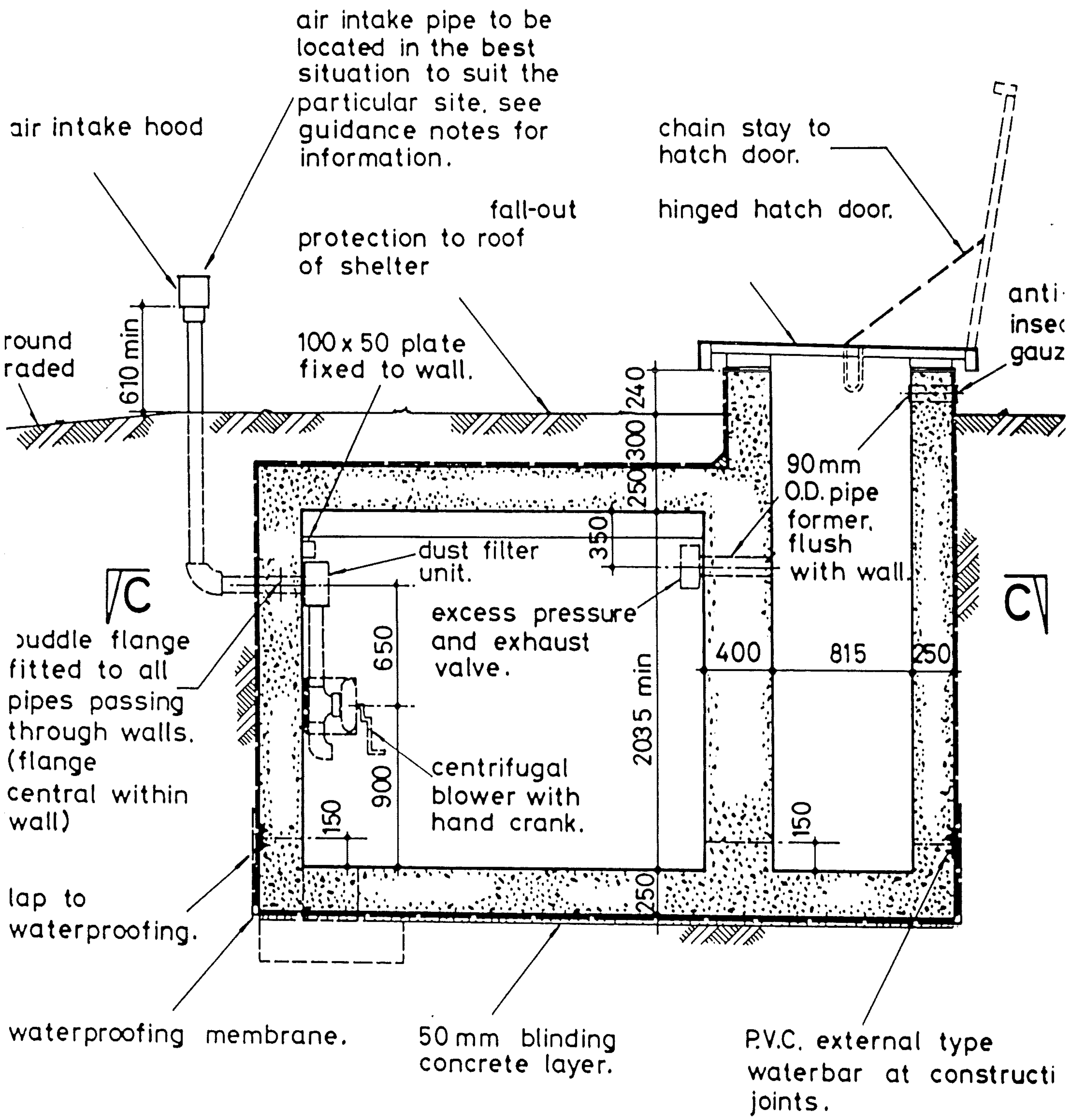


Figure E.1 Section through shelter (11).

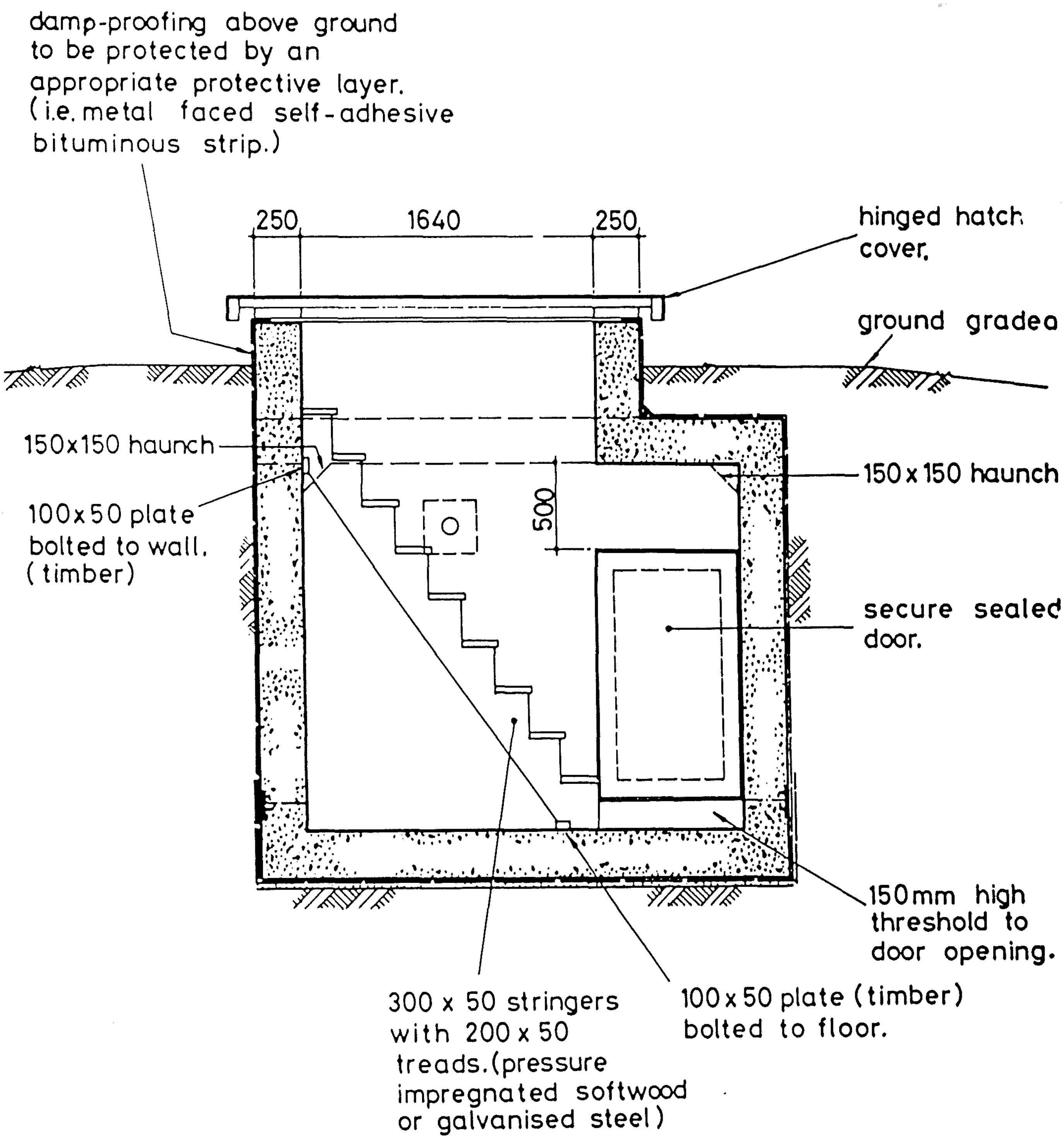


Figure E.2 Section through shelter (11).

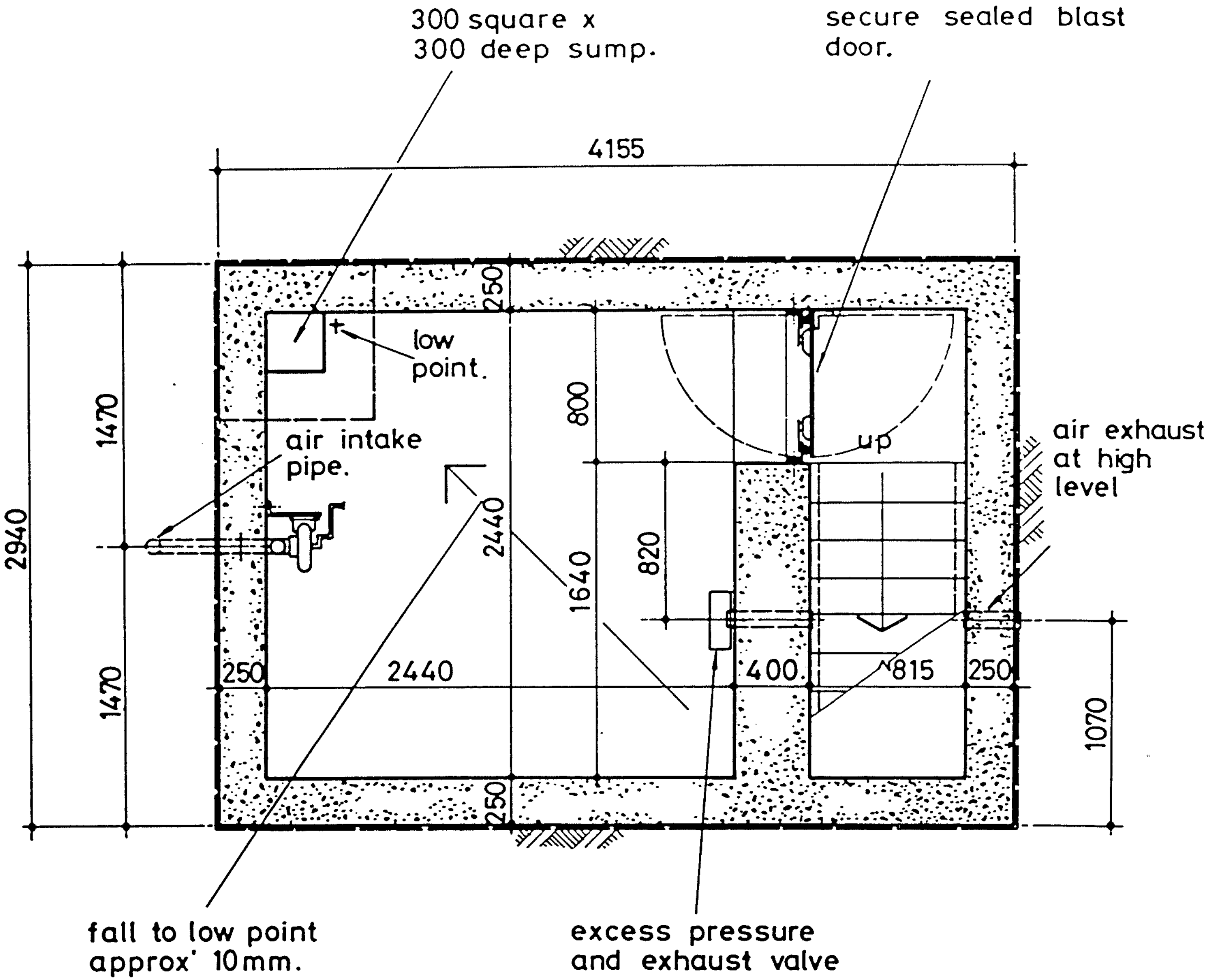


Figure E.3 Plan view of shelter (11).

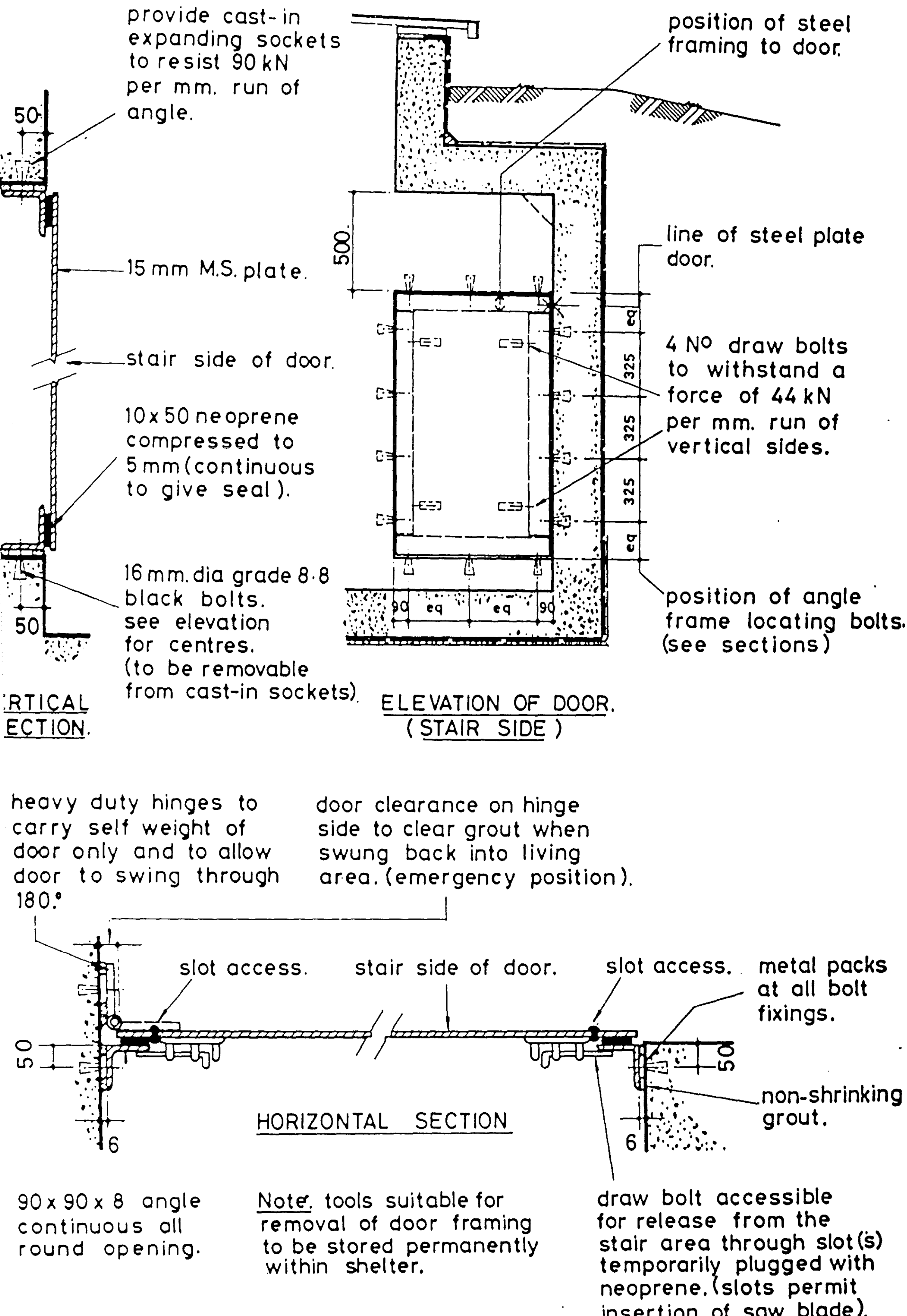


Figure E.4 Shelter detailing (11).

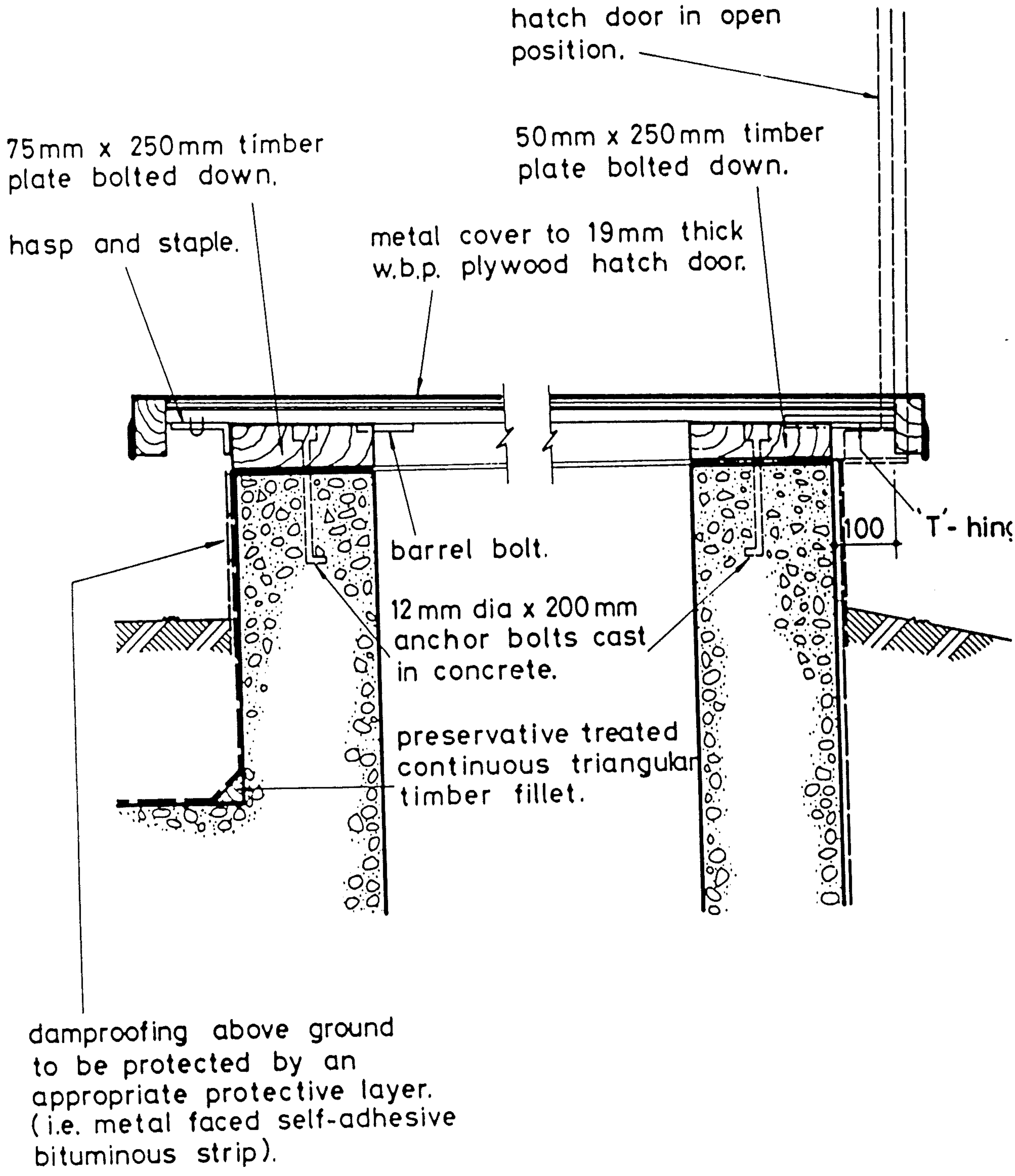
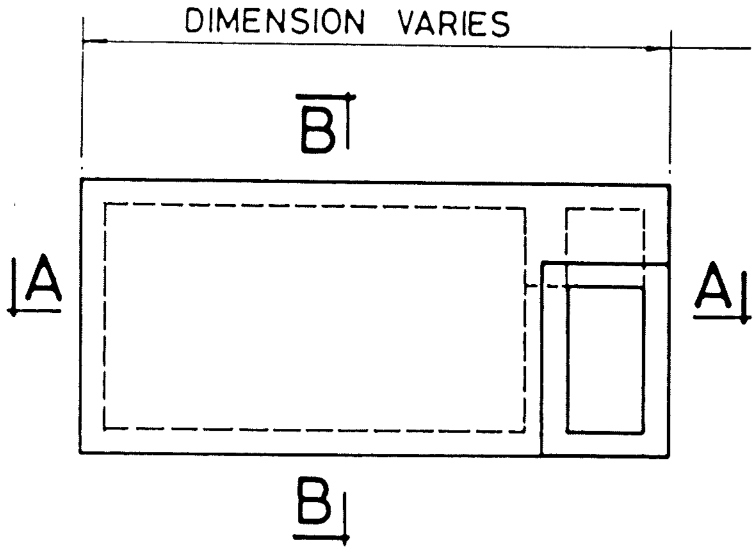
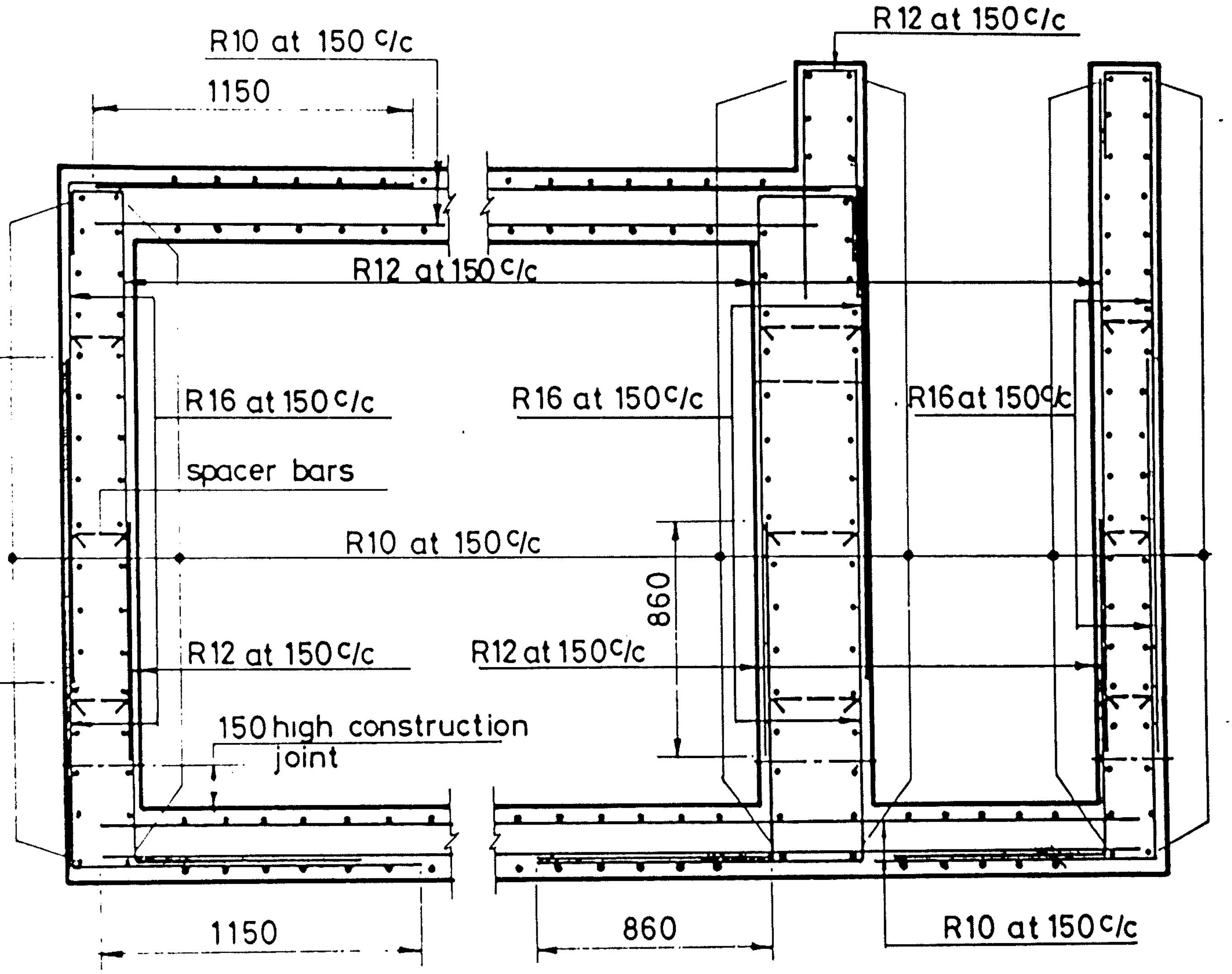


Figure E.5 Access hatch detail (11).



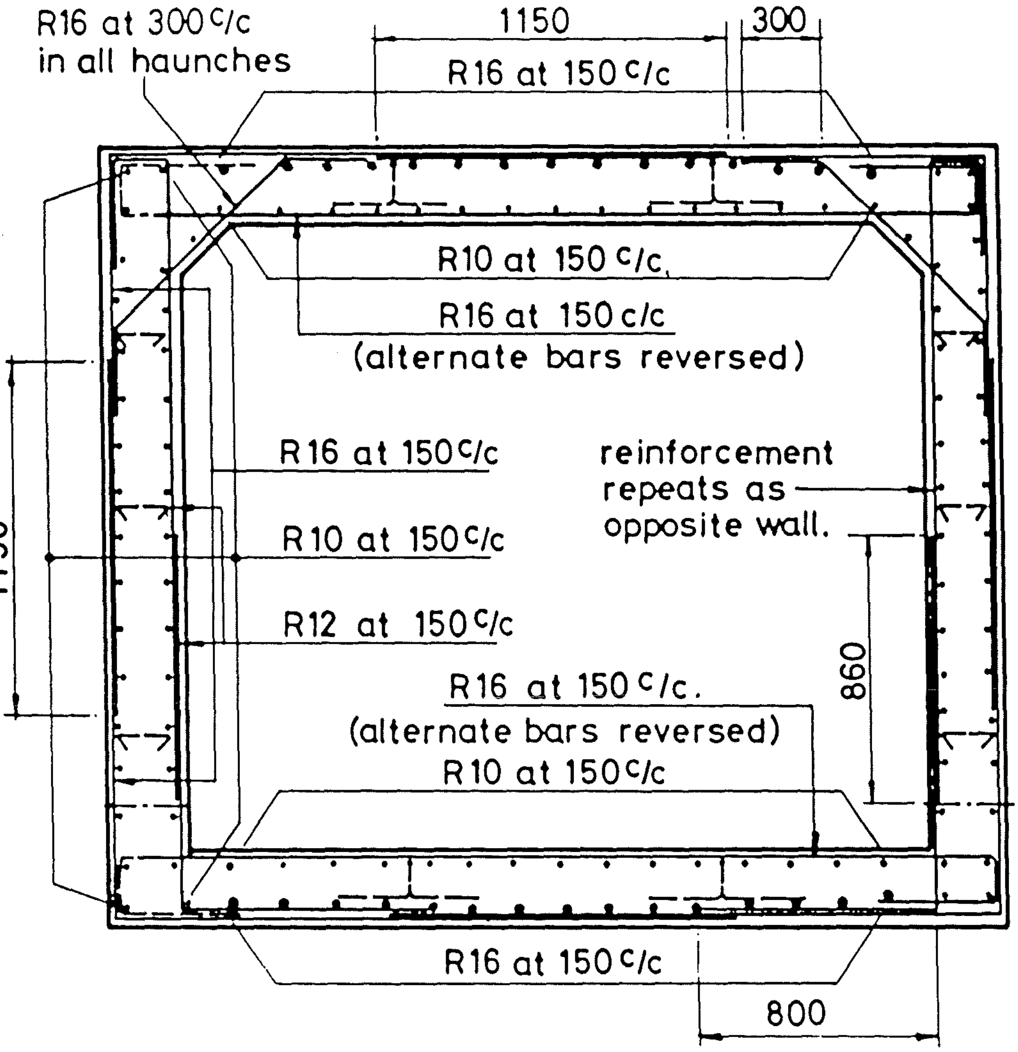
Note. Maintain bar dia' and spacings of reinforcement shown on sections for the length of shelter chosen.

PLAN OF 6, 8, 10 OR 12 PERSON SHELTER.



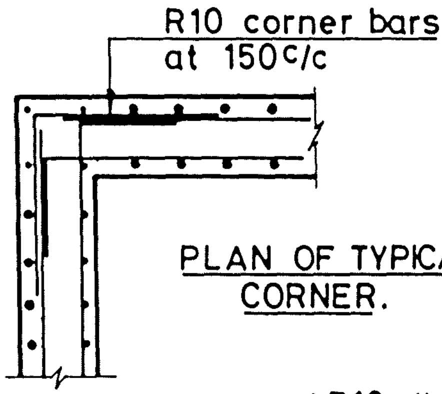
SECTION A-A

Figure E.6 Reinforced concrete detailing (11).

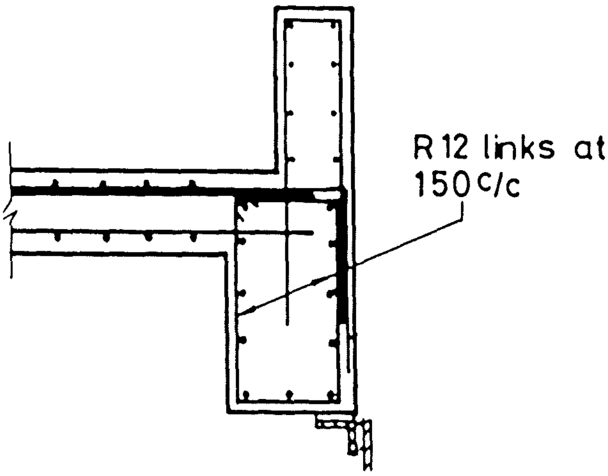


SECTION B-B

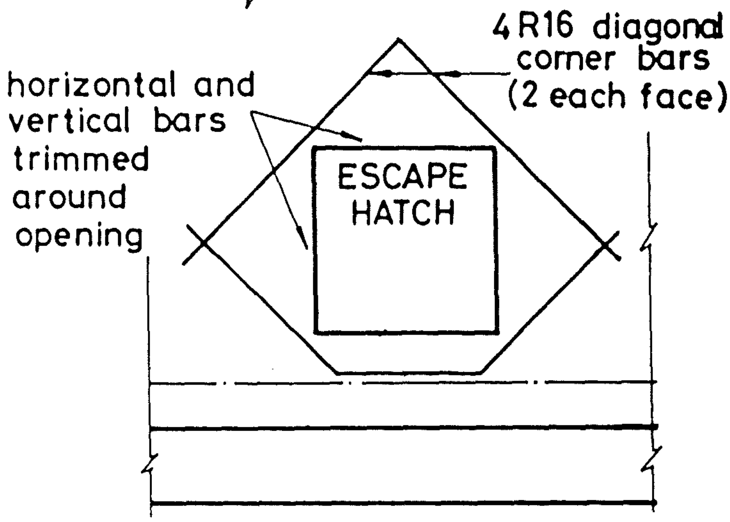
NOTES.
R16 = 16mm dia mild steel reinforcing rods.
Minimum cover to any reinforcement is to be 50mm.
An adequate number of spacer bars is to be provided between layers of reinforceme to maintain the required cover.



PLAN OF TYPICAL CORNER.



VERTICAL SECTION THROUGH DOWNSTAND BEAM OVER DOOR.



ELEVATION OF ESCAPE HATCH IN WALL.

Figure E.7 Reinforced concrete detailing (11).

APPENDIX F THE SEMILOOF SHELL ELEMENT BASIC THEORY

SYMBOLS

x, y, z	Local cartesian coordinates.
ξ, η	Curvilinear coordinates.
N	Membrane mapping function.
L	Rotational mapping function.

VECTOR NOTATION

$\bar{x}$, $\bar{y}$, $\bar{z}$	Unit vectors in local coordinates.
t	Scalar thickness.
T	Vector thickness.
$T\bar{z}$	Vector thickness resolute in z direction.
$R\bar{x}$	Vector thickness resolute in x direction.
$S\bar{y}$	Vector thickness resolute in y direction.

MATRIX NOTATION

[K]	Element stiffness matrix.
[B]	Shape function derivitives.
[D]	Element elasticity matrix.

DEFINITION OF TERMS

(See also Figs.F.1 and F.2)

DISPLACEMENTS:

Membrane:

u = Displacement in x direction.

v = Displacement in y direction.

Bending:

Wx = dw/dx = Rotation about y axis.

Wy = dw/dy = Rotation about x axis.

u = -z (dw/dx) = Displacement in x direction at height z due to rotations about the y axis.

v = -z (dw/dy) = Displacement in y direction at height z due to rotations about the x axis.

w = Displacement in z direction.

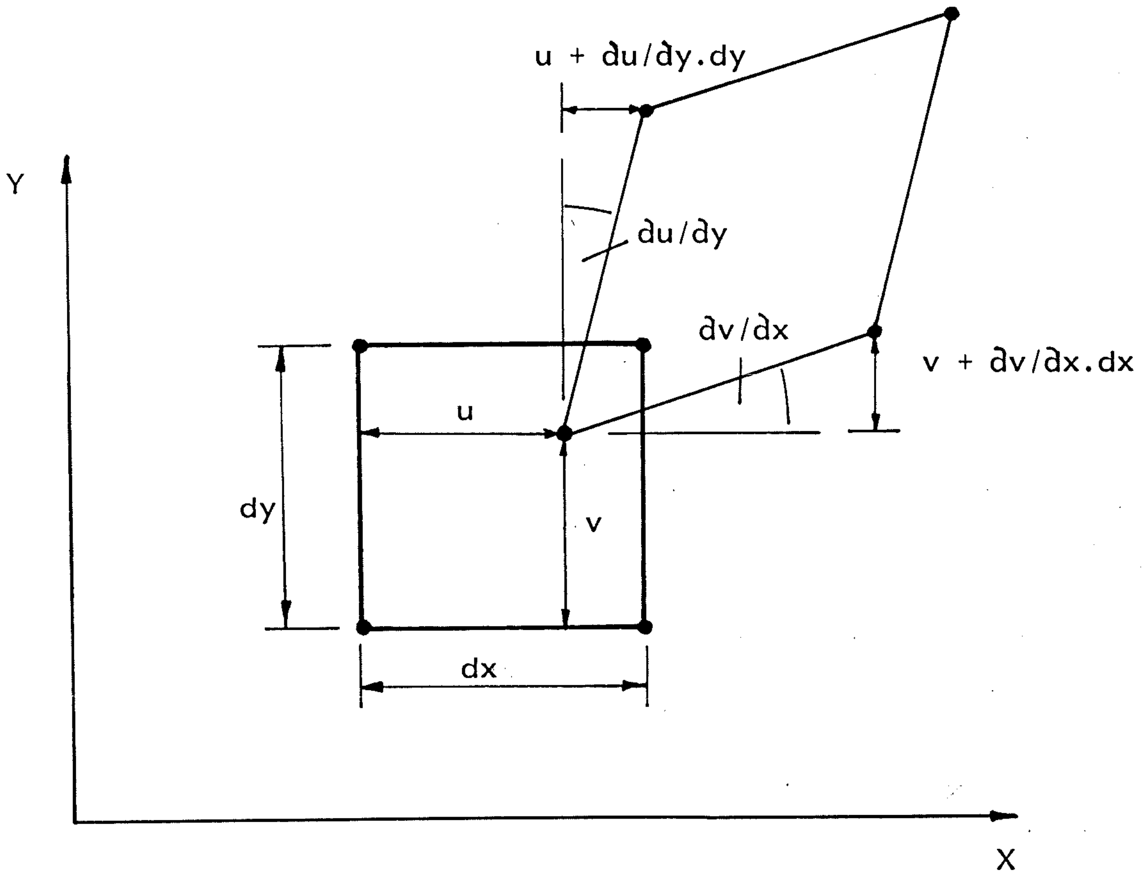


Figure F.1 Shear strain nomenclature.

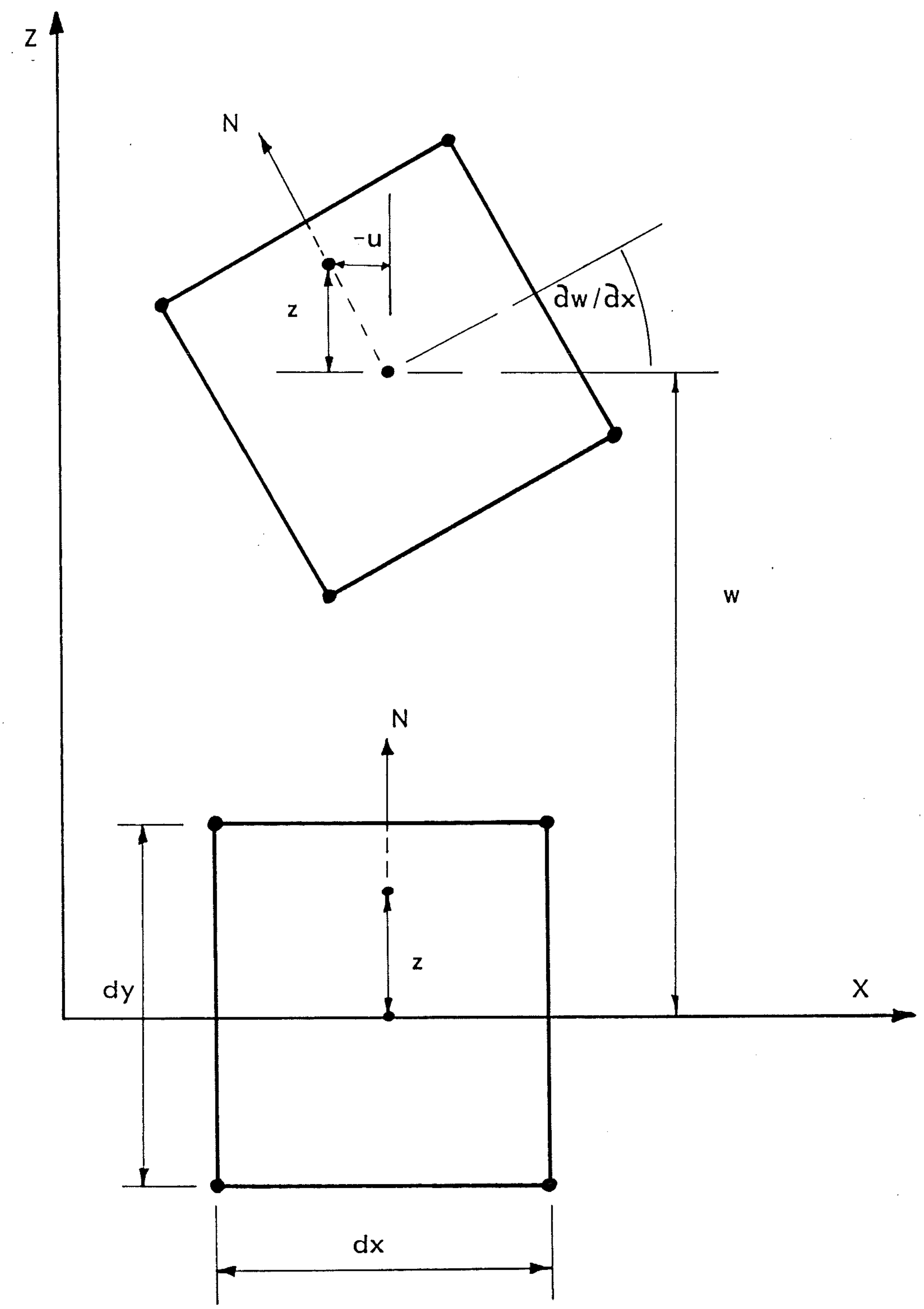


Figure F.2 Bending strain nomenclature.

STRAINS:

Membrane:

$Ux = du/dx$ = Direct tensile strain in the x direction.

$Vy = dv/dy$ = Direct tensile strain in the y direction.

$\gamma xy = (du/dy) + (dv/dx)$ = Shear strain in the xy plane.

Bending:

$Uxx = -z\ d/dx(Wx) = -z(Wxx) = -z(d^2w/dx^2)$

= Direct tensile strain in the x direction at height z, due to rotations about the y axis.

$Vyy = -z\ d/dy(Wy) = -z(Wyy) = -z(d^2w/dy^2)$

= Direct tensile strain in the y direction at height z, due to rotations about the x axis.

$\gamma xxyy = -z[\ d/dy(Wx) + d/dx(Wy)] = -z(Wxy + Wyx) = -2z(Wxy)$

$= -\ 2z(d^2w/(dxdy))$

= Shear strain in xy plane at height z, due to rotations about the x and y axes.

SemiLoof bending terms:

$Uxz = -Wxx = -d^2w/dx^2$ therefore; $z(Uxz) = Uxx$

$Vyz = -Wyy = -d^2w/dy^2$ therefore; $z(Vyz) = Vyy$

$Uyz + Vxz = -2Wxy = -2(d^2W/dxdy)$

therefore; $z(Uyz + Vxz) = \gamma xxyy$

STRESSES:

Membrane:

$\sigma x = [D]Ux = [D](du/dx)$ = Direct stress due to direct strain.

$\sigma y = [D]Vy = [D](dv/dy)$ = Direct stress due to direct strain.

$\tau xy = [D]\gamma xy = [D]((du/dy)+(dv/dx))$ = Shear stress due to shear strain.

Bending:

$\sigma xx = [D]Uxx = [D](-z(d^2w/dx^2))$ = Direct stress due to bending.

$\sigma yy = [D]Uyy = [D](-z(d^2w/dy^2))$ = Direct stress due to bending.

$\tau xxyy = [D]\gamma xxyy = [D](-2z(d^2w/dxdy))$ = Shear stress due to bending.

$Mx = t^3/(12z)\,\sigma xx$

$My = t^3/(12z)\,\sigma yy$

$Mxy = t^3/(12z)\,\tau xxyy$

F.1 Introduction

The SemiLoof shell element is a thin shell element, introduced in Section 1.2 and shown in Fig.1.1.

SemiLoof uses two sets of mapping functions as shown in Tables 1.1 and 1.2 and Figures 1.3 to 1.5. The N functions map the membrane responses and the L functions map the rotational responses, both with respect to the element curvilinear coordinate system (ξ,η). Local orthogonal coordinate sytems are also defined at the Loof nodes whose orientations are shown in Fig.1.2.

The stiffness matrix formulation is based on the volume integral:

$$[K] = \int_V [B]^t[D][B]\ dV \qquad (F.1)$$

where [B] = a matrix of differential operators of size d x n where d is the number of strain terms and n is the number of element degrees of freedom.

[D] = is the elasticity matrix which contains Young's modulus and Poisson ratio terms.

Stresses are calculated from:

$$[\sigma] = [D][B]\{u\} \qquad (F.2)$$

where {u} = a vector of displacement terms.

The size of the [B] matrix is determined by the number of element displacement terms (No. of equations) and the number of strain items for the theory.

The SemiLoof shell has 43 equations and calculates 6 strain terms.

The 43 equations are:

1. Those relating to translational displacement:

   8 nodes x 3 translations per node = 24 equations

2. Those relating to rotations:

   9 nodes x 2 rotations per node = 18 equations

3. An additional bubble function: = 1 equation

   ----

   43 equations

However, 11 equations are constrained out, leaving only 32 in the final formulation. Hence the final [B] matrix is of the order 6x32.

F.2 Strain formulations

The stress terms calculated from displacements interpolated at the standard 2x2 Gauss points are:

Tx = Force due to x strain from membrane action.

Ty = Force due to y strain from membrane action.

Txy = Shear force due to shear strain from membrane action.

Mx = Bending moment due to x strain from bending action.

My = Bending moment due to y strain from bending action.

Mxy = Bending moment due to xy strain from bending action.

where Tn = σn.Area

Mn = σnn.$t^2/6$

and n is the x, y or z direction.

The stress calculations are shown in Table F.1.

The strains required for thin shell theory are:

Ux = du/dx = Direct tensile strain in the x direction.

Vy = dv/dy = Direct tensile strain in the y direction.

γxy = (du/dy) + (dv/dx) = Shear strain in the xy plane.

Uxx = $-z(d^2w/dx^2)$ = Direct tensile strain in the x direction at height z, due to rotations about the y axis.

Vyy = $-z(d^2w/dy^2)$ = Direct tensile strain in the y direction at height z, due to rotations about the x axis.

γxxyy = $-2z(d^2w/(dxdy))$ = Shear strain in xy plane at height z, due to rotations about the x, y axes.

Table F.1 SemiLoof elasticty matrix and stress/strain terms

$$[\sigma] = [D][B]\{u\}$$

$$\begin{bmatrix} \sigma x \\ \sigma y \\ \tau xy \\ \sigma xx \\ \sigma yy \\ \tau xxyy \end{bmatrix} = \frac{Et}{(1-v^2)} \begin{bmatrix} 1 & v & 0 & 0 & 0 & 0 \\ v & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2(1-v) & 0 & 0 & 0 \\ 0 & 0 & 0 & t^2/12 & vt^2/12 & 0 \\ 0 & 0 & 0 & vt^2/12 & t^2/12 & 0 \\ 0 & 0 & 0 & 0 & 0 & t^2(1-v)/24 \end{bmatrix} \begin{bmatrix} Ux \\ Vy \\ \gamma xy \\ Uxx \\ Vyy \\ \gamma xxyy \end{bmatrix}$$

The strain terms are those which remain after the assumptions and constraints which define thin plate theory are imposed. Although the standard membrane strain terms are adopted in the SemiLoof shell, the rotational strain terms are derived independantly.

The total shear strain at the element surface due to bending and membrane action is:

$$du/dz = -z(d^2 w/dx^2) + du/dx$$
$$dv/dz = -z(d^2 w/dy^2) + dv/dy$$

F.4 SemiLoof strain terms

The SemiLoof shell basic theory is defined in equations F.3 and F.4.

F.4.1 Calculation of transverse shear strains

$$\begin{bmatrix} \gamma xz \\ \gamma yz \end{bmatrix} = \begin{bmatrix} Wx \\ Wy \end{bmatrix} + \frac{\delta}{T} \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} - \frac{1}{T} \begin{bmatrix} Ux & Uy \\ Vx & Vy \end{bmatrix} \begin{bmatrix} R \\ S \end{bmatrix} \qquad (F.3)$$

where

γxz = $du/dz + dw/dx$ = Shear strain on xz face

γyz = $dv/dz + dw/dy$ = Shear strain on xy face

Expanding equation F.3, for transverse shear strain, γxz:

$$\gamma xz = Wx + (\delta\bar{x} /T) - (R/T)Ux - (S/T)Uy$$

$$= dw/wx + (\delta\bar{x} /T) - (R/T)(du/dx) - (S/T)(du/dy)$$

but $\gamma xz = dw/dx + du/dz$

therfore; $du/dz = (\delta\bar{x} /T) - (R/T)(du/dx) - (S/T)(du/dy)$

where: $\delta\bar{x}$ is the difference in x displacement of the top and bottom surfaces.

The final term in equation F.3 is included to compensate for the geometry if points on the upper and lower surfaces are not directly above each other, i.e. not on a line normal to the neutral surface (Fig.F.3).

An orthogonal vector system is defined where R, S and T represent slope components of the vector thickness $\bar{T}$ (Fig.F.4):

$$\bar{T} = R\bar{x} + S\bar{y} + T\bar{z}$$

where:

T is the resolute thickness in the z direction.

R is the resolute thickness in the x direction.

S is the resolute thickness in the y direction.

$\bar{T}$ can be interpolated to the Gauss points within the element by using the Loof mapping functions:

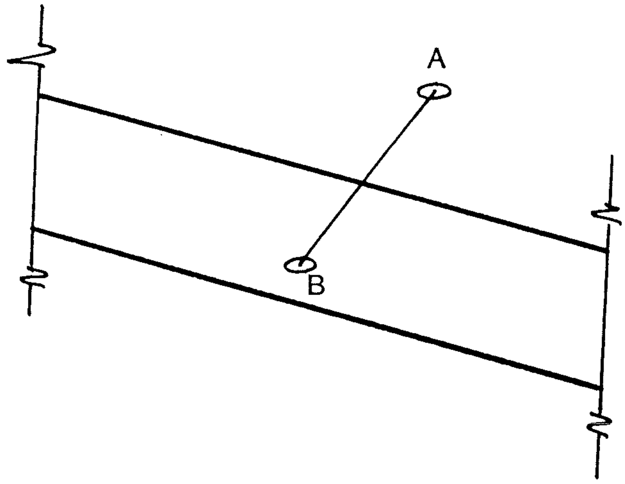


Figure F.3 Corrections are needed if A is not directly above B.

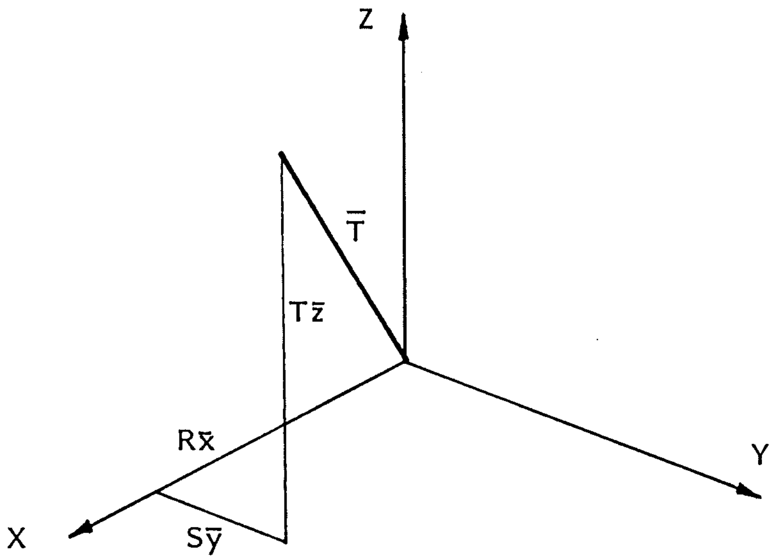


Figure F.4 Resolution of a thickness vector.

$$\bar{T} = \sum_{j=1}^{9} \bar{T}_j L_j$$

where Lj is a Loof mapping function.

F.4.2 Calculation of equivalent rotational second derivitives

One of the benifits of the SemiLoof formulation is that the need for the second derivitives of slope (Wxx) are not required, and equivalent terms like Uxz are calculated instead.

$$\begin{bmatrix} Uxz & Vxz \\ Uyz & Vyz \end{bmatrix} = \frac{1}{T}\begin{bmatrix} \delta x \\ \delta y \end{bmatrix}\begin{bmatrix} \bar{x} & \bar{y} \end{bmatrix} - \frac{1}{T}\begin{bmatrix} Rx & Sx \\ Ry & Sy \end{bmatrix}\begin{bmatrix} Ux & Vx \\ Uy & Vy \end{bmatrix} + \frac{1}{T}\begin{bmatrix} Tx \\ Ty \end{bmatrix}\begin{bmatrix} Wx & Wy \end{bmatrix} \quad \text{(F.4)}$$

where:

$Uxz = -Wxx = -d^2w/dx^2$

therefore;

z(Uxz) = Direct tensile strain in the y direction at height z, due to rotations about the x axis.

and:

$Vyz = -Wyy = -d^2w/dy^2$

therefore;

z(Vyz) = Direct tensile strain in the x direction at height z, due to rotations about the y axis.

and:

$$Uyz + Vxz = -2Wxy = -2(d^2W/dxdy)$$

therefore;

$$z(Uyz + Vxz) = \text{Shear strain in xy plane at height } z,$$

due to rotations about the x and y axes.

Expanding equation F.4 for Uxz:

$$Uxz = (\delta x\bar{x}\ /T) - (RxUx/T) - (SxVy/T) + (TxWx/T)$$

$$= (\delta x\bar{x}\ /T) - (Rx/T)(du/dx) - (Sx/T)(dv/dy) + (Tx/T)(dw/dx)$$

but; $Uxz = -Wxx = (-d^2w/dx^2) = (du/dx)(1/z)$

therefore:

$$(du/dx)(1/z) = (\delta x\bar{x}\ /T) - (Rx/T)(du/dx) - (Sx/T)(dv/dy) + (Tx/T)(dw/dx)$$

where: $\delta x\bar{x}$ is the difference in x displacement of the top and bottom surfaces, differentiated with respect to x.

$Tx = dT/dx$

$Rx = dR/dx$

$Sx = dS/dx$

The first term in equation F.4 gives the the strain, (Uxx/z). The following two terms are corrective terms.

The second term is included to account for a plate with varying nodal thicknesses. A vertical displacement in the neutral plane would have a horizontal component in the top and bottom surfaces and would therefore contribute to strain (Fig.F.5)

The third term accounts for the differences in surface area on the top and bottom surfaces of a deeply curved shell. This affects the area scaling factor between the two curvilinear coordinate systems which is the determinant of the Jacobian (Fig.F.6).

F.4.3 Shear constraints

Having defined the above strain terms, various constraints are implemented in order to satisfy the thin shell criteria.

The γyz shear strain is constrained at each of the Loof nodes.

$$\gamma yz = 0 \qquad = 8 \text{ constraints}$$

Both γxz and γxy shear strains at the central node are constrained by defining the expression:

$$\gamma = \gamma xz\bar{x} + \gamma yz\bar{y}$$

and integrating at the 2x2 Gauss points:

$$\int \gamma\bar{x} \; dA = \int \gamma\bar{y} \; dA = 0 \qquad = 2 \text{ constraints}$$

The vertical displacement at the central node is constrained out by setting the γxz shear around the perimeter to zero. The integration is carried out at the Loof nodes:

$$\int \gamma xz \; T \; d(\text{boundary}) = 0 \qquad = 1 \text{ constraint}$$

----

11 constraints

The constraints are imposed in the shell formulation by the use of a constraint matrix which reduces the original 43 equations to 32.

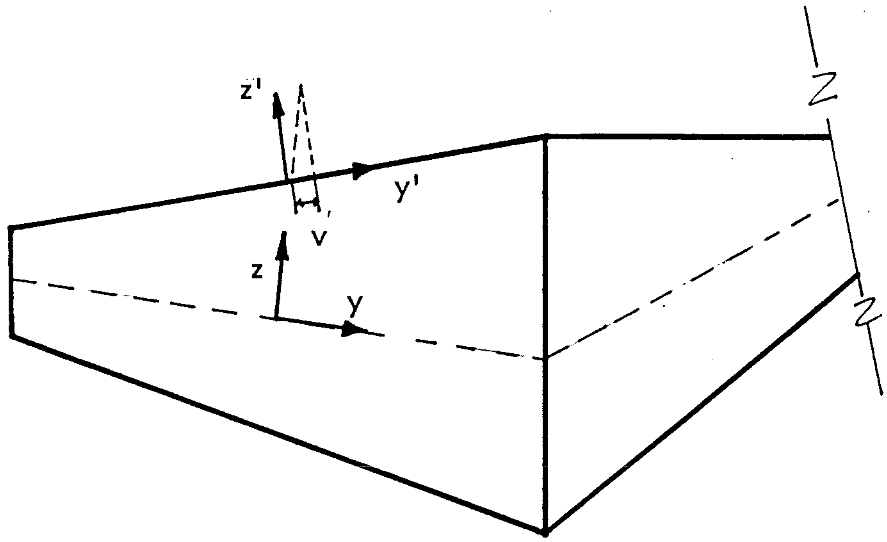


Figure F.5 Appearance of strain due to different nodal thicknesses.

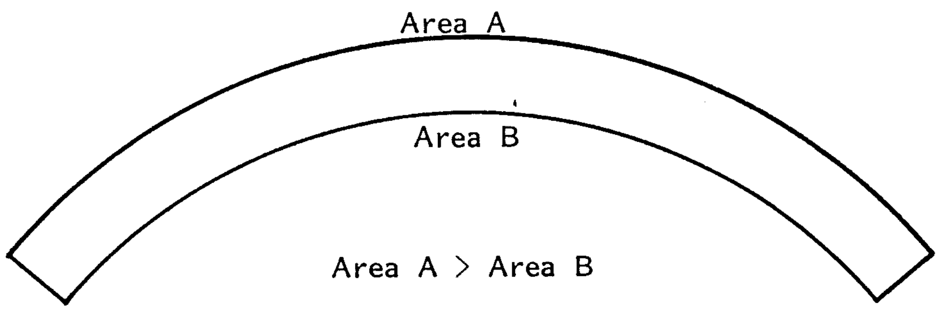


Figure F.6 Highly curved elements need corrections to the Jacobian.

The 32 equations which are left are:

u, v, w displacements at each corner and midside node.

= 24 equations

$\theta\bar{y}$ at each Loof node. = 8 equations

----

32 equations

Hence the strains required for thin shell theory can be calculated:

Membrane:

Ux = du/dx = Direct tensile strain in the x direction.

Vy = dv/dy = Direct tensile strain in the y direction.

γxy = (du/dy) + (dv/dx) = Shear strain in the xy plane.

Bending:

Uxx = -z d/dx(Wx) = -z(Wxx) = z(Uxz)

= Direct tensile strain in the x direction at height z, due to rotations about the y axis.

Vyy = -z d/dy(Wy) = -z(Wyy) = z(Vyz)

= Direct tensile strain in the y direction at height z, due to rotations about the x axis.

γxxyy = -z[d/dy(Wx) + d/dx(Wy)] = z(Uyz + Vxz)

= Shear strain in xy plane at height z, due to rotations about the x and y axes.